



Universidade Federal de Itajubá

INSTITUTO DE ENGENHARIA MECÂNICA

TRABALHO FINAL DE GRADUAÇÃO

Modelagem da detecção de danos via elementos finitos e redes neurais artificiais

Aluno: Marco Nieto Martínez

Matrícula7

Orientador: Prof^a. Patricia S. Lopes Alexandrino

2013

Dedicatória

Dedico este trabalho à minha querida família que sempre me apoiou e a todas as pessoas que dedicaram parte de seu tempo em me ajudar quando eu precisei.

Agradecimentos

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais presto minha homenagem:

Aos meus pais pelo incentivo em todos os momentos da minha vida.

A minha namorada pelo tempo e apoio que me deu.

À minha orientadora, que me mostrou os caminhos a serem seguidos.

A todos os professores e colegas, que ajudaram de forma direta ou indireta na conclusão deste trabalho.

*O verdadeiro sabor da vitória aparece naturalmente naqueles
que colaboraram desde a semente até a colheita*

Resumo

O trabalho tem por objetivo a modelagem do problema inverso de detecção de danos em estruturas por meio do método de elementos finitos e redes neurais artificiais. O método de elementos finitos é um método numérico aproximado para a resolução do problema direto de detecção de danos. As redes neurais artificiais é um método de identificação de parâmetros para a resolução do problema inverso. Deste modo, os princípios relacionados ao método de elementos finitos e às redes neurais artificiais serão estudados. Além disso, o aprendizado sobre como utilizar um software de elementos finitos (FEMAP/NASTRAN[®]) para obtenção dos dados necessários para o problema direto será necessário. O software MATLAB[®] será utilizado para a análise dos resultados obtidos via as redes neurais artificiais.

Palavras-chave: Detecção de danos, Elementos finitos, Redes neurais artificiais, FEMAP/NASTRAN[®], MATLAB[®].

Abstract

This work is all about the detection of damages by using software of finite elements and artificial neural networks. Principles of the finite elements and artificial neural networks have to be known, besides that learning how to work with this new software called (FEMAP/NASTRAN[®]). The purpose of the work is to obtain information after submitting the structure with charges and constrains, for there posterior analyze with the software (MATLAB[®]). This software will provide us with the ability of programming the artificial neural networks to certificate if the program have found where the holes where. Afterwards a discussion will be made about the results and the conclusions of the work done.

Keywords: Damage detection, Finite Elements, Artificial Neural Networks, FEMAP/NASTRAN[®], MATLAB[®]

Lista de Figuras

Figura 1 - Comparativa de análise de softwares (fonte: Soporte FEMAP & NX Nastran).....	16
Figura 2 - Seleção do arquivo de unidades do sistema internacional	18
Figura 3 - Seleção das unidades no programa	18
Figura 4 - Seleção da matéria	19
Figura 5 - Criação do sólido	20
Figura 6 - Seleção da posição do furo	20
Figura 7 - Peça furada.....	21
Figura 8 - Seleção do tipo de malha	21
Figura 9 - Zoom do furo com elementos maiores	22
Figura 10 - Zoom do furo com elementos menores	22
Figura 11 - Peça malhada	23
Figura 12 - Malha com elementos quadrados de uma camada.....	24
Figura 13 - Malha com elementos quadrados de três camadas	24
Figura 14 - Malha com elementos quadrados de uma camada com dados.....	25
Figura 15 - Malha com elementos quadrados de três camadas com dados	25
Figura 16 - Colocação das restrições	26
Figura 17 - Colocação do carregamento.....	27
Figura 18 - Peça restrita e carregada	28
Figura 19 - Análise da peça	29
Figura 20 - Peça analisada mediante as tensões de VonMises	29
Figura 21 - Diferentes análises possíveis	30
Figura 22 - Peça deformada.....	30
Figura 23 - Peça analisada e cortada, para ver as tensões no furo.....	31
Figura 24 - Zoom da peça cortada no furo	32
Figura 25 - Localização dos furos e dos sensores	33

Figura 26 - Zoom do furo com os dados	34
Figura 27 - Função sigmoidal.....	38
Figura 28 - Esquema modelo neural.....	38
Figura 29 - Esquema das camadas da rede neuronal	40
Figura 30 - Esquema da janela do processo	47
Figura 31 - Novos dados obtidos e furos numerados	49
Figura 32 - Teste para o Furo 1 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda.....	51
Figura 33 - Teste para o Furo 2 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda.....	52
Figura 34 - Teste para o Furo 3 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda.....	53
Figura 35 - Teste para o Furo 4 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda.....	54
Figura 36 - Teste para o Furo 5 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda.....	55
Figura 37 - Esquema de peça furada submetida a tração uniaxial.....	60
Figura 38 - Zoom no furo	62
Figura 39 - Zoom no furo deformado	62
Figura 40 - Zoom dados de teste de VonMisses.....	64

Lista de Tabelas

Tabela 1 - Dados tomados dos testes das peças de R=15mm.....	33
Tabela 2 - Dados tomados dos testes das peças de R=20mm.....	34
Tabela 3 - Novos dados tomados das peças de R=15 mm.	50
Tabela 4 - Novos dados tomados das peças de R=20 mm.	50
Tabela 5 - Dados dos erros para o Furo 1	52
Tabela 6 - Dados dos erros para o Furo 2.....	53
Tabela 7 - Dados dos erros para o Furo 3.....	54
Tabela 8 - Dados dos erros para o Furo 4.....	55
Tabela 9 - Dados dos erros para o Furo 5.....	56

Lista de Abreviaturas e Siglas

FEMAP - *Finit Element Modeling And Postprocesing*

NASTRAN - *NAsa STructural ANalysis Program*

CAD - *Computer Aided Design*

MATLAB - *MAtrix LABoratory*

ESP - *Enterprise Software Porducts Inc*

SDRC - *Structural Dynamics Research Corporation*

MSC - *MacNeal-Schwendler Corporation*

FAR - *Federal Acquisition Regulation*

AI - *Artificial Intelligence*

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1 Conceitos Fundamentais.....	1
1.2 Motivações e objetivos	1
1.3 Revisão da literatura	2
1.4 Estruturas do trabalho.....	2
CAPÍTULO 2 - ESTUDO DO MÉTODO DE ELEMENTOS FINITOS	4
2.1 Introdução sobre o método dos elementos finitos	4
2.2 Historia do método dos elementos finitos	4
2.3 Conceitos gerais do método.....	5
2.4 Equações gerais do método dos elementos finitos	8
2.4.1 Campo das deformações.....	8
2.4.2 Deformações unitárias.....	9
2.4.3 Estado de tensões	10
2.5 Passos a seguir para calcular a solução do problema de elementos finitos	12
2.6 Elasticidade bidimensional	13
2.7 Elasticidade tridimensional	14
CAPÍTULO 3 - ESTUDO DO PROGRAMA FEMAP/NASTRAN® E PROCESO DE MODELAGEM E ANÁLISE.....	15
3.1 História do FEMAP/NASTRAN®	15
3.1.1 FEMAP®	15
3.1.2 NASTRAN®	17
3.2 Processos de fabricação da peça a ser analisada.....	17
3.2.1 Configuração do <i>software</i> FEMAP®	17
3.2.2 Determinação das propriedades e o material da peça	19
3.2.3 Desenho da peça no FEMAP®	19
3.2.4 Criação da malha da peça.....	21
3.2.5 Aplicação das condições de contorno	25

3.2.6 Análise da peça mediante NASTRAN®	28
3.2.7 Pós-processamento dos dados	29
3.2.8 Tomada de dados.....	32
CAPÍTULO 4 - PRINCÍPIOS FUNDAMENTAIS DAS REDES NEURAIS ARTIFICIAIS	35
4.1 Introdução.....	35
4.2 Revisão Histórica.....	35
4.3 Conceito de rede neuronal e vantagens	36
4.4 Modelos neurais.....	37
4.5 Processo de aprendizagem.....	40
CAPÍTULO 5 - ESTUDO DO PROGRAMA MATLAB	45
5.1 História do MATLAB	45
5.2 Processo de utilização do software MATLAB® para implementação das redes neurais.	45
CAPÍTULO 6 - RESULTADOS E DISCUSSÃO	49
6.1 Resultados.....	49
CAPÍTULO 7 - CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS.....	57
7.1 Conclusões.....	57
7.2 Sugestões para trabalhos futuros	57
REFERÊNCIAS BIBLIOGRÁFICAS	58
APÊNDICE A - Conceito de tensão numa placa com furo.....	60
APÊNDICE B - Critério de <i>Von Misses</i>	63

CAPÍTULO 1 - INTRODUÇÃO

1.1 Conceitos Fundamentais

Este trabalho consiste na modelagem do problema inverso de detecção de danos via elementos finitos e redes neurais artificiais. O método de elementos finitos está implementado em um software chamado FEMAP/NASTRAN[®] que é utilizado para a modelagem da peça mediante o programa de CAD. O software também é utilizado para posteriores análises com um número determinado de furos na peça.

Após a análise realizada no software, dados das diferentes tensões que se localizam em distintas posições da peça são coletados para sua posterior utilização dentro de um programa implementado com o software MATLAB[®]. Dentro deste software está desenvolvido um programa (*toolbox*) baseado em princípios das redes neurais artificiais. Com este programa já feito, para dadas dimensões (raio) dos furos e para os dados das tensões obtidos do programa FEMAP/NASTRAN[®] vai se tentar, através do programa, determinar a localização dos furos na placa, com a intenção de poder utilizá-los como um localizador de pequenos defeitos. Estes defeitos podem ocorrer dentro de diferentes materiais em diversos campos da engenharia. Assemelha-se a um método de inspeção não destrutiva e prevenção de possíveis anomalias dentro da estrutura, que poderiam causar falhas em condições futuras, mediante a tomada de dados de diversos pontos de um material.

Este trabalho está particularizado para furos, e aplicado sobre elementos básicos, como no caso das placas retangulares, já que as capacidades do aluno em desenho com programas de CAD não são muito elevadas.

1.2 Motivações e objetivos

Para a realização deste trabalho não havia nenhum modelo prévio estudado para melhorar ou desenvolver. Este trabalho foi feito devido à existência de um código sobre redes neurais desenvolvido no programa MATLAB[®]. Então surgiu a possibilidade de começar a trabalhar com um *software* (FEMAP/NASTRAN[®]) bastante utilizado dentro do setor de

cálculo de estruturas. O conhecimento do funcionamento deste *software* pode ser útil em futuras aplicações relacionadas com diversos campos da engenharia.

É importante ressaltar que para a realização deste trabalho, o aluno não conhecia o programa FEMAP/NASTRAN[®]. Então, foi um desafio conhecê-lo e estudá-lo o suficiente para poder fazer uma estrutura simples e os cálculos necessários para a resolução do trabalho.

1.3 Revisão da literatura

Na verdade não foram utilizados muitos artigos para a coleta de informações, já que, até agora foi feito mais um trabalho prático que teórico. Foi preciso consultar alguns livros para o desenvolvimento do capítulo dois, já que a teoria dos elementos finitos evolui muito desde sua criação. Estas consultas são citadas naquele capítulo.

Mas além de ter utilizado estes livros baseados na teoria dos elementos finitos, foram consultados as ajudas (*helps*) e tutoriais disponíveis no programa (FEMAP/NASTRAN[®]). Isso porque ainda não existem livros e artigos conhecidos que explicam como utilizá-lo.

Consultas de artigos e livros foram feitas para a revisão da literatura das redes neurais, foram escolhidos e utilizaram-se principalmente os autores Serrano, Izaurieta e Saavedra , e Lopes. Para a utilização do programa MATLAB[®], foram utilizados ajudas disponíveis na ajuda (*help*) do programa e conhecimentos prévios do aluno em relação a esse programa.

1.4 Estruturas do trabalho

O trabalho foi estruturado começando pela introdução sobre o trabalho apresentada no Capítulo 1. Depois no Capítulo 2 apresentando o método dos elementos finitos e discorrendo sobre as equações que o regem, a fim de estabelecer uma melhor compreensão de como funcionará o *software*, já que a coluna principal dele são os elementos finitos. O Capítulo 3 apresenta o *software* FEMAP/NASTRAN[®], detalhando-se um passo a passo desde como se faz a peça no *software*, passando pelas aplicações das condições de contorno (restrições e carregamento), e posteriormente apresentando uma análise dos resultados. Este capítulo está complementado com imagens do *software*.

No Capítulo 4 é apresentada a rede neural artificial, quais são suas equações principais e como discorre seu funcionamento, já que esta é a base do código implementado no programa MATLAB[®]. No Capítulo 5 tem uma breve historia do programa utilizado é como foi desenvolvida a utilização do software para os testes. A posterior no Capitulo 6 se

apresentam os resultados obtidos nos testes de uma forma numérica mediante tabelas de dados, e de uma forma visual, já que se têm imagens das localizações dos furos reais e os achados pelo programa. Para concluir tem-se o Capítulo 8 onde estão as conclusões do trabalho e as diferentes possibilidades `para trabalhos futuros.

Também foram feitos dois apêndices necessários para uma melhor compreensão de alguns dos aspectos dentro do capítulo dois, acerca das tensões que se experimentam dentro de uma placa furada e sobre os critérios de *Von Misses*.

CAPÍTULO 2 - ESTUDO DO MÉTODO DE ELEMENTOS FINITOS

2.1 Introdução sobre o método dos elementos finitos

O método dos elementos finitos adquiriu uma importância muito relevante na resolução de problemas tanto físicos como de engenharia, porque mediante este método é possível resolver problemas que até pouco tempo atrás, somente eram possíveis de resolver com métodos matemáticos tradicionais. Além disso, esses problemas precisavam também de processos que consistiam na realização de um protótipo que seguia uma sequência de ensaios iterativos para determinar as possíveis melhoras, até alcançar o resultado esperado (HUEBNER, 1975). Estes tipos de processos se traduzem em custos de produção muito elevados, e em um longo tempo de desenvolvimento.

Entretanto com o desenvolvimento de novos sistemas informáticos, estes processos de elementos finitos sofreram uma grande evolução, que possibilitou a resolução destes problemas mediante a realização de um modelo matemático de cálculo do sistema real. Tal modelo é mais fácil e mais econômico de modificar que o protótipo, mas, além disso, ainda não é um modelo aproximado de cálculo (HUEBNER, 1975).

A estrutura básica do método dos elementos finitos já era conhecida há muito tempo, mas sua formulação matemática é relativamente nova. O desenvolvimento dos sistemas informáticos facilitou sua aplicação a usuários com diferentes tipos de programas para a resolução de problemas, como no caso deste trabalho e o programa FEMAP/NASTRAN®.

2.2 Historia do método dos elementos finitos

O método dos elementos finitos surgiu no ano de 1960 quando foi utilizado por Clough (1960) em um documento que falava sobre os problemas da elasticidade plana. Além disso, o problema dos elementos finitos foi referenciado muito antes dessa data, porque tanto os matemáticos, como os físicos e os engenheiros começaram a utilizar essas ideias para resolver os problemas que tinham. Os matemáticos precisavam desse método para resolver problemas de contorno, os físicos queriam aproximar funções para representá-las contínuas e

os engenheiros tentavam procurar um jeito de conseguir reforçar as estruturas e a influência da rigidez nelas.

Couran (1943) propôs a utilização das funções polinômicas para a formulação de problemas elásticos em regiões triangulares. Depois dele, Greenstadt (1959) realiza a aproximação mediante a discretização dos elementos em celas, melhor que em pontos. O que aconteceu foi que ele reduziu um domínio contínuo a um domínio discreto. A solução atualmente conhecida como o método dos elementos finitos foi publicada por Turner *et al.* (1956). Nesta solução, os autores introduziram as aplicações de elementos finitos simples, barras e placas triangulares, e utilizaram os conceitos das funções de forma e discretização do problema. Zienkiewicz e Cheung (1965) fizeram uma interpretação do método dos elementos finitos e falaram de sua possível aplicação em qualquer problema de campos. Nesse documento eles mostram que as equações dos elementos finitos podem ser obtidas utilizando métodos de aproximação de pesos residuais, tal como o método Galerkin ou mínimos quadrados. Este avanço fez com que os matemáticos se interessassem muito no método dos elementos finitos, isso porque se interessavam no processo de resolução das equações diferenciais lineares e não lineares.

Hoje em dia o método dos elementos finitos é uma das mais potentes ferramentas de cálculo para a resolução de problemas da engenharia e ciência. O método está expandindo-se devido ao desenvolvimento de computadores que resolvem as diferentes equações dos elementos finitos.

2.3 Conceitos gerais do método

A ideia principal do método dos elementos finitos é a divisão de um sistema contínuo em um conjunto de elementos de um tamanho menor. Estes elementos ficam conectados entre si por uma série de pontos que chamamos nós. Isto é denominado discretização dos corpos.

A equação que governa o comportamento do sistema contínuo também será a que governa os elementos. Dessa forma o que este método faz é passar de um sistema com infinitos graus de liberdade (contínuo) a um sistema já discretizado, que terá finitos graus, e poderá ser resolvido por um sistema de equações, sejam lineares ou não (COOK, 1995; REDDY, 2006).

Em qualquer sistema que tenha que ser analisado, pode-se distinguir uma série de aspectos fundamentais, são eles:

- *Domínio*: É o espaço onde será analisado o sistema;
- *Condições de contorno*: São as variáveis conhecidas que condicionam o sistema de uma forma determinada, como as restrições, os carregamentos, as temperaturas, etc;
- *Incógnitas*: São as variáveis do sistema que precisam ser conhecidas depois que as condições de contorno tenham sido aplicadas no sistema, podem ser tensões deformações, temperaturas, etc.

Hipóteses da discretização: Numa estrutura discreta, a sua deformação vem definida por um número finito de parâmetros, todos eles juntos conformam o vetor das deformações $\{\Delta\}$. Mas num meio contínuo a estrutura tem infinitas formas de se deformar. Estas formas são independentes umas de outras. Por enquanto a configuração da deformada não pode vir como um vetor com um número finito de elementos, como o vetor $\{\Delta\}$. Deve ser uma função vetorial $\{u\}$ que nos indica qual que é a deformação de qualquer ponto da estrutura, e fica formada por três componentes escalares (Equação (1)).

$$u = \begin{Bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{Bmatrix} \quad (1)$$

Em geral esta função $\{u\}$ será a solução do problema, e se o problema está bem planejado terá que cumprir as condições de contorno. Mas não se pode ter certeza de que esta função tenha uma expressão analítica, ou de que exista algum jeito de achá-la (REDDY, 2006; LIZARZA, 2008).

O método dos elementos finitos utilizará a hipótese de discretização para tentar resolver este problema. Os principais conceitos do método são os seguintes:

- O meio contínuo é dividido em pequenas regiões próximas entre si e com formas geométricas simples. Estas formas geométricas estão normalizadas e são chamadas de elementos finitos. Os elementos finitos ficam próximos entre eles num número finito de pontos denominados nós;
- Os deslocamentos desses nós são as incógnitas fundamentais do problema. O deslocamento de um ponto será determinado pelos deslocamentos dos nós dos elementos ao qual o ponto pertence;

- As funções de interpolação são as que permitirão achar os deslocamentos interiores por meio de interpolação dos deslocamentos ocorridos nos nós;
- As funções de interpolação e os deslocamentos nodais são as que definirão o estado das deformações do elemento. Como é conhecido o estado das deformações, mediante a equação constitutiva do material que relacionam ambos estados, ficará definido o estado das tensões.

Esta hipótese da discretização é uma das ferramentas mais importantes dentro do corpo dos elementos finitos.

Outro conceito importante mencionado anteriormente são as funções de interpolação (OTTOSEN, 1992; LIZARZA, 2008). Para compreendê-las veja um exemplo: Seja um elemento finito no plano definido por um número determinado de nós n , este ponto vai ter um vetor de deslocamentos $\{u\}$ (Equação (2)), e os graus de liberdade dos nós do elemento formarão o vetor $\{\delta\}$ (Equação (3)).

$$u = \begin{Bmatrix} u(x, y) \\ v(x, y) \end{Bmatrix} \quad (2)$$

$$\delta = \{U_1 V_1 U_2 V_2 \dots U_n V_n\} \quad (3)$$

Então para aproximar o campo de deformações, as funções de interpolação (N_i) serão utilizadas (Equação (4) e Equação (5)).

$$u = \sum N_i U_i \quad (4)$$

$$v = \sum N_i V_i \quad (5)$$

As funções de interpolação (N_i) são utilizadas tanto para os deslocamentos dos elementos na direção x (U), como na direção y (V). Os dois deslocamentos são interpolados separadamente.

Existe outro jeito de expressar a interpolação das deformações, que é o modo matricial. Este modo relaciona o vetor dos deslocamentos com o vetor dos graus de liberdade mediante a matriz das funções de interpolação (HUEBNER, 1975; LIZARZA, 2008). Esta matriz vai ter tantas filas quantos deslocamentos tenham o ponto determinado, e tantas colunas quantos graus de liberdade tenham o elemento (Equação (6)). Para os problemas gerais da elasticidade este é o aspecto que tem a matriz.

$$N = \left(\begin{array}{cc|cc|ccc} N_1 & 0 & N_2 & 0 & \dots & 0 & N_n & 0 \\ 0 & N_1 & 0 & N_2 & 0 & \dots & 0 & N_n \end{array} \right) \quad (6)$$

2.4 Equações gerais do método dos elementos finitos

Dentro do conceito das equações gerais do método, as equações principais do método serão explicadas.

2.4.1 Campo das deformações

O campo das deformações, como foi escrito anteriormente, está definido, para qualquer ponto, pelo vetor das deformações $\{u\}$. Este vetor terá tantas componentes quantas deformações existam no domínio (LIZARZA, 2008).

Mediante a hipótese da interpolação, qualquer campo de elementos finitos pode ser aproximado ao campo das deformações, fazendo uso das funções de interpolação como os fatores de ponderação das deformações de cada nó de cada elemento. Tendo como vetor $\{\delta\}$ o vetor das deformações nodais de todos os elementos, pode-se escrever de forma matricial a equação do vetor das deformações conforme a Equação(7).

$$u = \begin{pmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{pmatrix} \quad (7)$$

Adicionando as Equações (4) e (5) à Equação (8)

$$w = \sum N_i W_i \quad (8)$$

E complementando a Equação (3) para que fique como a Equação (9)

$$\delta = \{U_1 V_1 W_1 U_2 V_2 W_2 \dots U_n V_n W_n\} \quad (9)$$

Logo, o vetor deformações fica como a Equação (10)

$$u = N \delta \quad (10)$$

Resta ainda definir como a matriz das funções de interpolação deve ser. Esta matriz terá três filas, conforme as três dimensões do espaço, e terá tantas colunas quantos graus de liberdade tenham os nós do elemento Equação(11).

$$N = \left[\begin{array}{ccc|ccc} N_1 & 0 & 0 & N_n & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_n & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_n \end{array} \right] \quad (11)$$

2.4.2 Deformações unitárias

Para passar das deformações de um ponto determinado do domínio às deformações unitárias precisa-se de um operador matricial que as relacione. Haverá neste operador um número determinado de filas, que será o número de deformações que tenha o elemento, e tantas colunas quantos componentes tenha o campo de deslocamentos (LIZARZA, 2008). Vejam-se as Equações (12) e (13)

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \partial u \quad (12)$$

$$\varepsilon = \partial u = \partial N \delta \quad (13)$$

Nesta relação (Equação(13)) se identifica a matriz B (Equação(14)).

$$B = \partial N \quad (14)$$

Então se cumprirá a seguinte relação (Equação(15)).

$$\varepsilon = B \delta \quad (15)$$

Agora pode-se escrever o problema de uma maneira diferente, onde a matriz B vai fazer o relacionamento das deformações unitárias e o vetor das deformações nodais em um ponto qualquer do elemento. Então o que a matriz B esta representando é o campo das deformações unitárias que fica no interior do elemento finito, como resultado da utilização da hipótese da interpolação.

Seguindo a Equação (14) pode-se expressar a matriz B utilizando-se a estrutura da matriz N e aplicando-se o operador matricial (Equação (16) e (17)).

$$B = \partial N = \partial N = \begin{bmatrix} N_1 & 0 & 0 & \dots & N_n & 0 & 0 \\ 0 & N_1 & 0 & \dots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \dots & 0 & 0 & N_n \end{bmatrix} \quad (16)$$

$$B = [B_1 B_2 \dots B_n] \quad (17)$$

Onde cada matriz B tem a forma mostrada na Equação (18)

$$B = \partial \begin{bmatrix} N_1 & 0 & 0 \\ 0 & N_1 & 0 \\ 0 & 0 & N_1 \end{bmatrix} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix} \quad (18)$$

2.4.3 Estado de tensões

As tensões de qualquer ponto do domínio vão ficar definidas mediante o tensor de tensões (Equação (19)). Além disso, as equações constitutivas do material são responsáveis pelo relacionamento das tensões com as deformações (Equação (20)).

$$\sigma = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{Bmatrix} \quad (19)$$

$$\sigma = D(\varepsilon - \varepsilon_0) + \sigma_0 \quad (20)$$

Onde se supõe que a matriz D tem valores constantes que dependerão das propriedades intrínsecas do material, como o módulo de elasticidade (E) e o coeficiente de Poisson (ν). Junto com esta matriz D aparecem os valores das deformações e tensões iniciais do material (REDDY, 2006; LIZARZA, 2008).

Através de uma pequena conclusão pode se deduzir que o problema dos elementos finitos reside principalmente em tentar achar os deslocamentos dos nós que formam os elementos finitos, porque ao conseguir achar estes deslocamentos e utilizando as funções de interpolação podem-se achar as deformações, e utilizando as equações constitutivas podem-se encontrar as tensões.

2.4.4 Equações de equilíbrio

Para as aplicações no cálculo estrutural as equações de equilíbrio podem ser encontradas minimizando a energia potencial do sistema que tem uma expressão conforme a Equação (21):

$$\pi = \int_V [\sigma]^T \varepsilon dV - \int_V [\delta]^T p dV - \int_S [\sigma]^T q dS \quad (21)$$

Sendo:

- σ - Vetor de tensões;
- ε - Vetor de deformações;
- δ - Deslocamentos em qualquer ponto;
- p - Forças de unidade de volume;
- q - Forças superficiais.

As integrais dos dois primeiros termos da equação são tomadas sobre a totalidade do volume da estrutura, mas no último termo a integral é tomada sobre a superfície carregada. O primeiro termo da direita da igualdade refere-se à energia da deformação interna da estrutura, o segundo e terceiro termo, respectivamente, são as contribuições de trabalho realizadas pelas forças de volume e superfície.

A energia potencial do meio contínuo será a soma das energias potenciais dos elementos finitos que o formam (Equação (22)).

$$\pi = \sum_e \pi_e \quad (22)$$

Podendo expressar a equação da energia de um modo diferente, substituindo algumas equações, chega-se na expressão dada pela Equação(23).

$$\frac{\partial \pi}{\partial \delta} = \frac{1}{2} \int_v ([B]^T DB) \delta dV - \int_v [N]^T p dV - \int_s [N]^T q dS \quad (23)$$

Mas pode-se também expressar a minimização da energia de um modo mais compacto (Equação (24)).

$$\frac{\partial \pi}{\partial \delta} = K \delta - F \quad (24)$$

Sendo:

- F - Forças nodais em cada elemento;
- K - Matriz de rigidez de cada elemento.

Fazendo a relação destas duas últimas equações (Equação (23) e (24)) pode-se observar como ficam as equações das forças nodais e da matriz de rigidez. (LIZARZA, 2008) Para minimizar a energia potencial o que se deve fazer é igualar a zero a equação derivada da energia potencial. Como resultado será obtido um sistema um sistema de equações de equilíbrio para todo o meio contínuo e, ao resolver este sistema, os deslocamentos nos nós serão encontrados.

2.5 Passos a seguir para calcular a solução do problema de elementos finitos

Para se obter a solução do problema de elementos finitos, metodicamente os seguintes pontos podem ser seguidos (COOK, 1995; OTTOSEN, 1992):

- Divisão do meio contínuo em elementos finitos;
- Avaliação da rigidez de cada um dos elementos e dos pontos de carregamento sobre o meio contínuo;
- Montagem dos elementos e das cargas;
- Resolução do sistema de equações lineais para obter os deslocamentos dos nós;
- Obtenção das deformações e tensões através dos deslocamentos dos nós.

2.6 Elasticidade bidimensional

Os problemas de elasticidade bidimensional são muito frequentes no ambiente da engenharia. Neste caso o meio contínuo ficará no plano XY onde se considera a espessura desprezível. A posição do ponto ficará definida somente por duas coordenadas (x,y) e sua deformação também terá somente duas componentes.

As equações que governam o comportamento do sistema em duas dimensões são as equações gerais particularizadas no caso 2-D (LIZARZA, 2008) Onde as funções de interpolação são as mesmas e podem se expressar em sua forma matricial como mostrado na Equação (25), Equação (3) e Equação (6).

$$u = N \delta \quad (25)$$

Podem-se expressar também as deformações unitárias $\{\varepsilon\}$ e a relação delas com as deformações $\{u\}$ mediante a expressão dada na Equação (26).

$$\varepsilon = \partial u = \partial N \delta = B \delta \quad (26)$$

A matriz B que representa o campo de deformações unitárias no interior do elemento finito e pode ser expressa como mostrado na Equação(27).

$$B = \partial N \left[\begin{array}{c|c|c|c} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} \\ \hline \vdots & \vdots & \vdots & \vdots \\ \frac{\partial N_n}{\partial x} & 0 & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_n}{\partial y} & 0 & \frac{\partial N_n}{\partial y} \\ \hline \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} \end{array} \right] \quad (27)$$

Na continuação, pode-se observar o estado de tensões em três dimensões de acordo com a Equação(28) que corresponde à equação constitutiva que relaciona as deformações com as tensões mediante a matriz D (Equação(29)).

$$\sigma = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{Bmatrix} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} \quad (28)$$

$$D = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \quad (29)$$

Onde λ é dada pela Equação (30) e μ é dada pela Equação (31).

$$\lambda = \frac{Ev}{(1+\nu)(1-2\nu)} \quad (30)$$

$$\mu = \frac{E}{2(1+\nu)} \quad (31)$$

A expressão da matriz D é a expressão para o caso tridimensional, já que para o caso bidimensional terá duas formas diferentes, dependendo se o problema é de elasticidade plana ou de tensão plana. A matriz terá que ser particularizada com as condições apropriadas para cada problema (LIZARZA, 2008).

2.7 Elasticidade tridimensional

Os conceitos da elasticidade tridimensional são os mesmos que os da elasticidade bidimensional, já que as equações que controlam os problemas são similares as que governam o problema bidimensional. Mas neste caso, uma terceira coordenada z e um terceiro deslocamento w tem que ser incluídos.

As equações do caso tridimensional são as mesmas que as equações mostradas na seção das equações gerais, já que as equações para cada tipo de dimensão podem ser obtidas a partir delas (LIZARZA, 2008). Tendo algumas variações em função das dimensões como a matriz D que neste caso vai ser uma matriz 6x6 e terá a mesma forma que a matriz D dada na Equação(29).

CAPÍTULO 3 - ESTUDO DO PROGRAMA FEMAP/NASTRAN® E PROCESSO DE MODELAGEM E ANÁLISE

3.1 História do FEMAP/NASTRAN®

NASTRAN® (*N*asa *S*tructural *A*nalysis *P*rogram) com FEMAP® (*F*init *E*lement *M*odeling *A*nd *P*ostprocessing) formam um conjunto que combina dois dos programas mais potentes e famosos de cálculo estrutural e térmico do mundo, pelo método dos elementos finitos, devido às capacidades de Pré/Pós-processamento do FEMAP® e aos diferentes módulos de análises por elementos finitos de NASTRAN® (FEMAP - Real FEA made easy).

3.1.1 FEMAP®

O programa FEMAP® foi lançado no ano de 1985 pela companhia *Enterprise Software Products Inc* (ESP) como um programa de Windows e somente como um pré-processador. Em Setembro de 1999, a companhia ESP foi adquirida por SDRC (*S*tructural *D*ynamics *R*esearch *C*orporation). No começo do ano de 2005 anunciaram o lançamento de uma atualização do FEMAP®, a versão V9.0 que inclui também uma integração com NASTRAN® e capacidades mais avançadas de resolução de equações não lineais. Hoje em dia o FEMAP® pode ser utilizado tanto como pré-processador ou pós-processador (FEMAP - Real FEA made easy; Soporte FEMAP & NX Nastran).

As distintas fases do programa são:

- *Pré-processador*: o FEMAP® tem diferentes capacidades como a criação da geometria do problema que vai ser estudado ou como a possibilidade de importar a geometria dos programas específicos de CAD, como por exemplo, do *Autocad*, *SolidEdge*, *CatiaV5* etc. O FEMAP® também tem ferramentas importantes para a discretização da geometria a estudar, que consiste em dividir a geometria em elementos finitos. Pode ser feito manualmente, semiautomático ou automático. Uma grande quantidade de condições de contorno e restrições pode ser introduzida. Além das condições de carregamento para poder desenhar os testes do melhor jeito possível, também é possível acessar a biblioteca

do programa para carregar as propriedades dos materiais que já estão pré-carregadas, ou o usuário pode utilizar o material que desejar para fazer o teste. Além disso, diferentes tipos de análises podem ser realizados, como análises estáticas, de deformações modais, transmissão do calor, etc.

A Figura 1 mostra os tipos de análises que é capaz de pós-processar o FEMAP[®] para diferentes pacotes de métodos de elementos finitos. Neste trabalho, somente é de interesse a primeira coluna do NX NASTRAN[®].

Análise	NX Nastran	MSC Nastran	Abaqud	Ansys	LS Dyna	Marc
Estático	•	•	•	•		•
Modal	•	•	•	•		•
Flambagem	•	•	•	•		
Transferência de calor	•	•	•	•		
Não linear	•	•	•	•	•	•
Resposta transitória	•	•	•	•	•	
Resposta em frequência	•	•	•	•		
Resposta aleatória	•	•		•		
Dinâmica explícita			•		•	
Pós-processado	•	•	•	•		•

Figura 1 - Comparativa de análise de softwares (fonte: Soporte FEMAP & NX Nastran)

- *Pós-processador:* Após o processo de análises ser realizado, existem diferentes ferramentas para analisar os resultados. Passando pelas visualizações gráficas de deformações, tensões, energias, pelos processos de animação onde o usuário pode ver a peça se mover devido às condições de contorno e aos carregamentos impostos. Além das

ferramentas numéricas que proporcionam informações (FEMAP - Real FEA made easy;,,Soporte FEMAP & NX Nastran).

3.1.2 NASTRAN®

O programa NASTRAN® foi criado no ano de 1965 como um programa para resolver elementos finitos. Foi desenvolvido pela NASA e financiado pelo o governo dos EUA para realizar cálculos das estruturas em projetos espaciais e aeronáuticos. Em 1976 a NASA vendeu seu código a diferentes empresas, conseguindo assim um desenvolvimento do código e da interface muito mais rápido. Hoje em dia as empresas mais importantes que comercializam NASTRAN® são MSC (MSC NASTRAN®) e UGS PLM Solutions (NX NASTRAN®) que é o *software* usado neste projeto (Soporte FEMAP & NX Nastran).

3.2 Processos de fabricação da peça a ser analisada

Em primeiro lugar, a peça deve ser designada e, posteriormente estudada. Esta peça é uma peça retangular de dimensões 200x400x10mm com diferentes tamanhos de furos em suas superfícies.

Nesta Seção 3.2, o processo detalhado desde o começo da realização da peça, passando pela criação dos furos, até a análise com elementos finitos mediante NASTRAN®, será ilustrado.

3.2.1 Configuração do *software* FEMAP®

Antes de começar a utilizar o programa para a realização dos testes, o *software* tem que ser configurado dentro das unidades corretas para o teste. O software vem configurado com unidades que não são as do sistema internacional, então isso será mudado para possibilitar seu uso. O arquivo deve ser carregado na janela do material (Figura 2) para que o *software* fique configurado de forma adequada. Este arquivo se encontra dentro da pasta de instalação do programa (Soporte FEMAP & NX Nastran; FEMAP – Finite Element Modeling...).

- Caminho para alterar as unidades: *File > Preferences > Library/Startup*

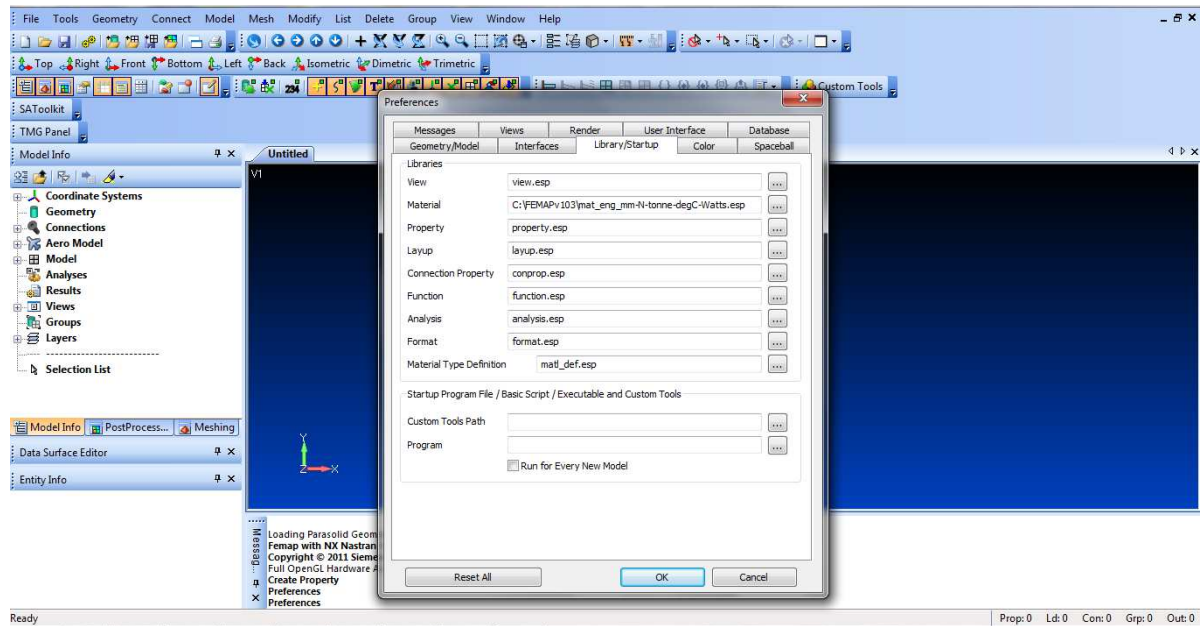


Figura 2 - Seleção do arquivo de unidades do sistema internacional

Como a peça tem um tamanho determinado de 200x400x10mm, o sistema de unidades será configurado em milímetros. O programa, por padrão, está em polegadas. As opções na aba (*Geometry/Model*) devem ser mudadas para a unidade desejada, ao ser milímetros terá um fator de escala de mil (Figura 3).

- Caminho: *File > Preferences > Geometry/Model*

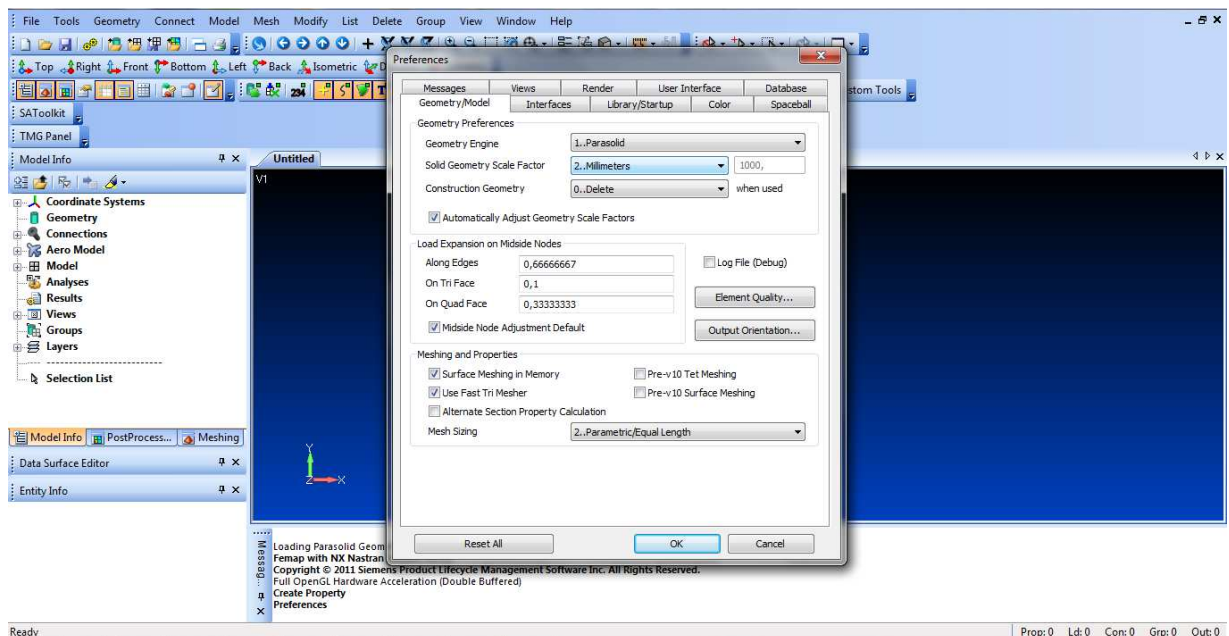


Figura 3 - Seleção das unidades no programa

3.2.2 Determinação das propriedades e o material da peça

O primeiro passo para começar a fazer o teste é determinar dentro do programa que tipo de material irá ser utilizado para a realização do ensaio. O material escolhido foi um alumínio do tipo 7075-T6. Os dados deste material que o programa precisa para a realização do ensaio se encontram carregados na base de memória do programa. Tabelas ou *data sheet* de dados do alumínio foram pesquisados para conferir que os valores não diferiam muito (*Aluminum 7075 Heat Treated (T6) Wrought*). Também foi determinado que o ensaio deveria ser realizado com um formato de peça tipo placa. Como pode ser apreciado na Figura 4, as unidades das propriedades do material (módulo de elasticidade e máxima tensão) estão expressas em MPa.

- Caminho: *Model > Property > Property/Type*

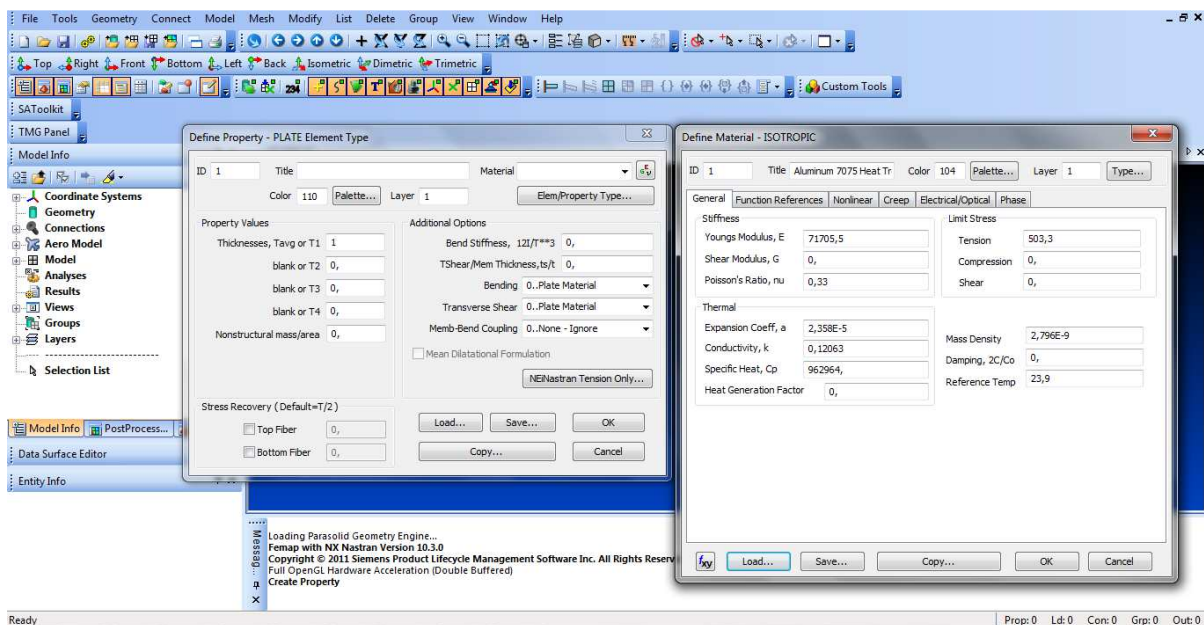


Figura 4 - Seleção da matéria

3.2.3 Desenho da peça no FEMAP®

Neste ponto, será mostrado o processo pelo qual a peça foi desenhada (FEMAP – Finite Element Modeling...).

- I. Criação de um sólido de dimensões 200x400x10mm (Figura 5).

- Caminho: *Geometry/Solids/Primitives*

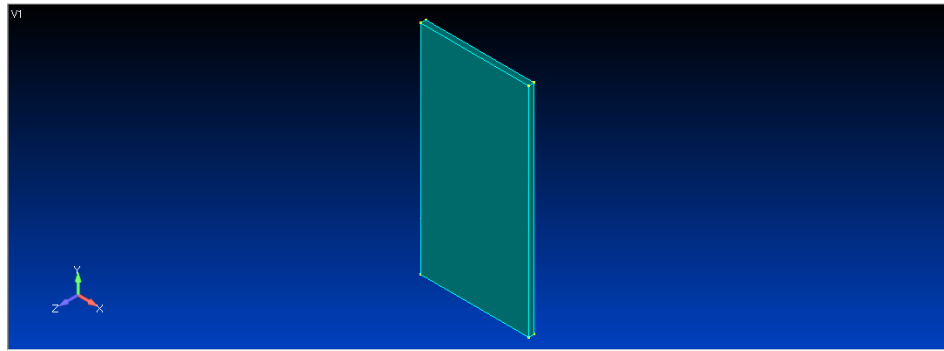


Figura 5 - Criação do sólido

II. Realização de um furo: Na Figura 6 é mostrado como foi realizado o furo na parte central da peça. A Figura 7 mostra o furo feito.

- Caminho: *Geometry/Solids/Extrude*

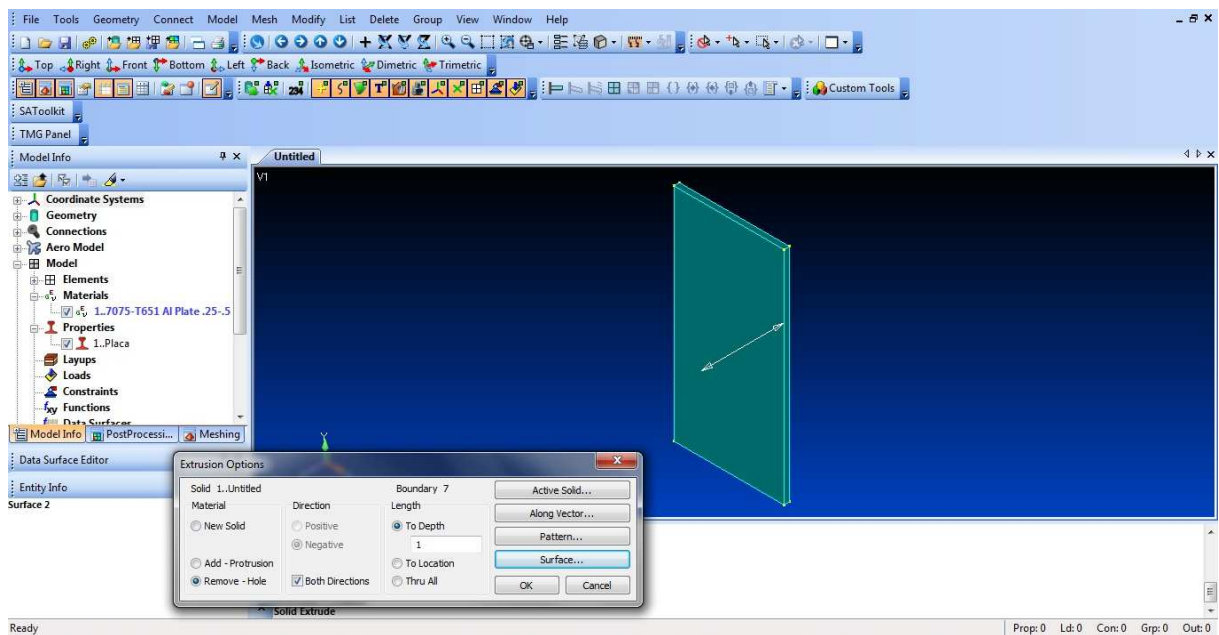


Figura 6 - Seleção da posição do furo

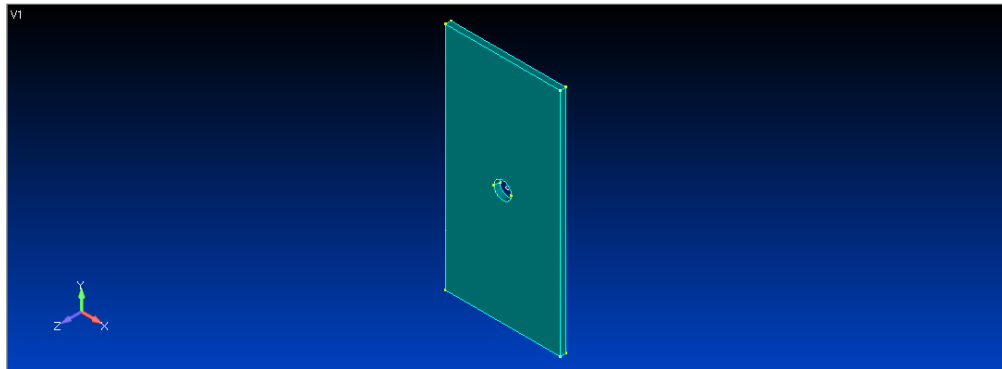


Figura 7 - Peça furada

3.2.4 Criação da malha da peça

Para poder realizar a simulação numérica deve-se definir uma malha para todas as superfícies da peça. Esta malha está definida por uma série de nós. Neste programa existe a possibilidade de selecionar qual tipo de elementos terá a malha. Pode-se escolher entre elementos do tipo quadrado, ou do tipo triangular. Os elementos do tipo triangulares (Figura 8) foram selecionados porque para superfícies furadas observou-se que elementos triangulares se ajustam melhor para o refinamento da malha (FEMAP – Finite Element Modeling...).

- Caminho: *Mesh/Geometry/Surface*

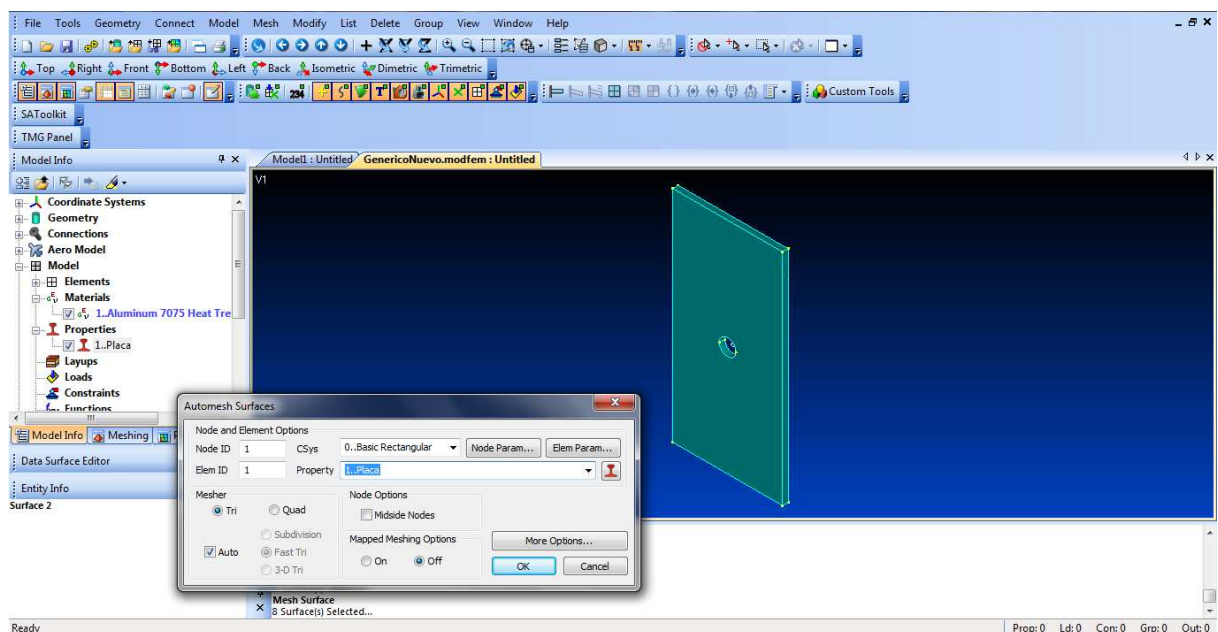


Figura 8 - Seleção do tipo de malha

Mas o refinamento da malha não depende apenas do tipo de elemento do qual ela é feita (quadrado ou triangular). O tamanho dos elementos com os que a peça será malhada também deve ser levado em conta. Uma ótima relação entre o tamanho da peça e o tamanho dos elementos precisa ser encontrada para que os resultados sejam os mais parecidos possíveis com a realidade. As Figuras 9 e 10 mostram um *zoom* na região do furo mostrando o uso de elementos de tamanho maior e menor, respectivamente.

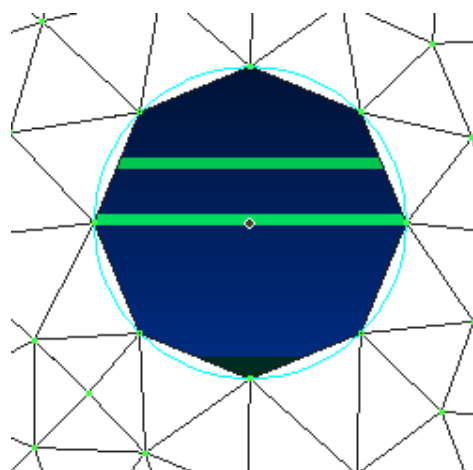


Figura 9 - Zoom do furo com elementos maiores

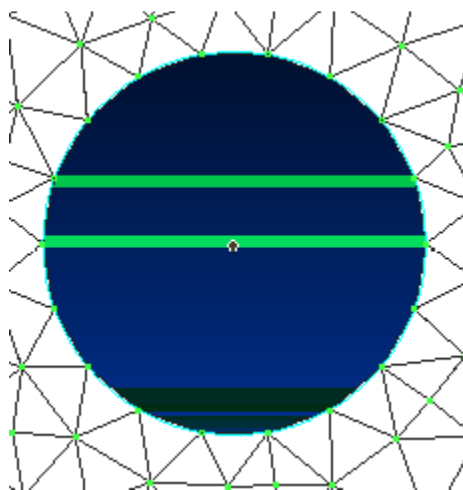


Figura 10 - Zoom do furo com elementos menores

Nestas duas figuras observam-se como os elementos menores ajustam-se melhor ao furo, e, portanto se ajustará melhor às análises. Entretanto o sistema tem um poder de cálculo determinado, e quanto menor sejam os elementos, mais complicado será para realizar os

cálculos e resolver as equações. O critério escolhido para selecionar o tamanho dos elementos foi o que melhor ajustava o furo sem atrapalhar a capacidade de cálculo do computador, foi um critério pessoal e uma sugestão a futuras modificações já que não foi seguido nenhuma base. Deve-se encontrar um equilíbrio entre estes fatores.

Depois de determinar tais aspectos, falta somente realizar o processo de malhar a peça determinada. Na Figura 11 pode-se observar a peça malhada.

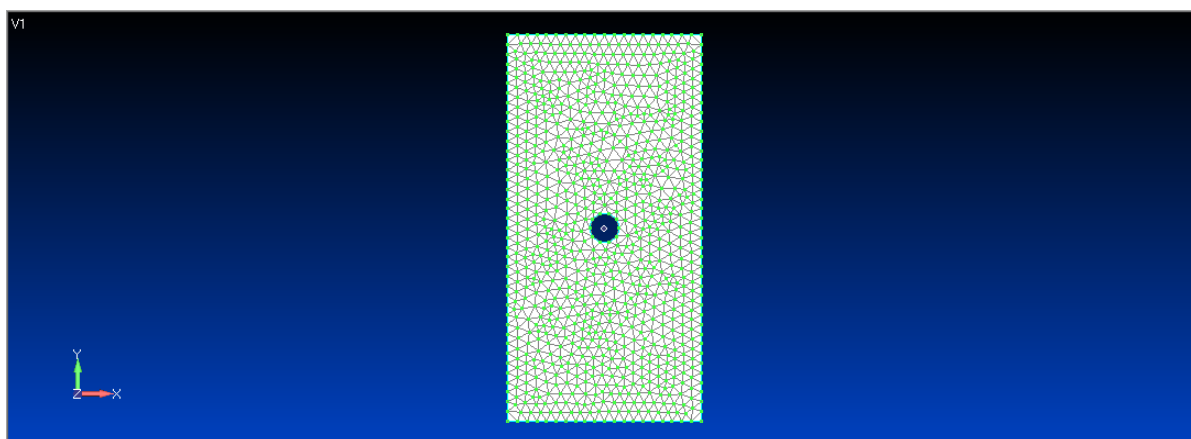


Figura 11 - Peça malhada

Apesar de que, para a realização do trabalho foi selecionado o elemento triangular para malhar a peça, também serão descritos alguns aspectos do elemento quadrado. Malhar a peça por elementos quadrados desenha em toda a peça elementos quadrados do mesmo tamanho, e se tem a possibilidade de fazer diferentes camadas de elementos quadrados próximos das circunferências para que a transição de tensões, desde o interior do furo até a superfície, tenha um processo mais gradual. Posteriormente serão mostradas algumas imagens que denotam a diferença. Mesmo que, esta característica não se enquadre aos elementos triangulares, acredita-se que para placas furadas, os melhores resultados para as posteriores análises são os proporcionados por os elementos triangulares. Por isto estes foram escolhidos para a realização do teste.

As figuras abaixo exemplificam os testes realizados nas proximidades do furo. Um teste realizado com uma camada (Figura 12) e outro realizado com três camadas (Figura 13).

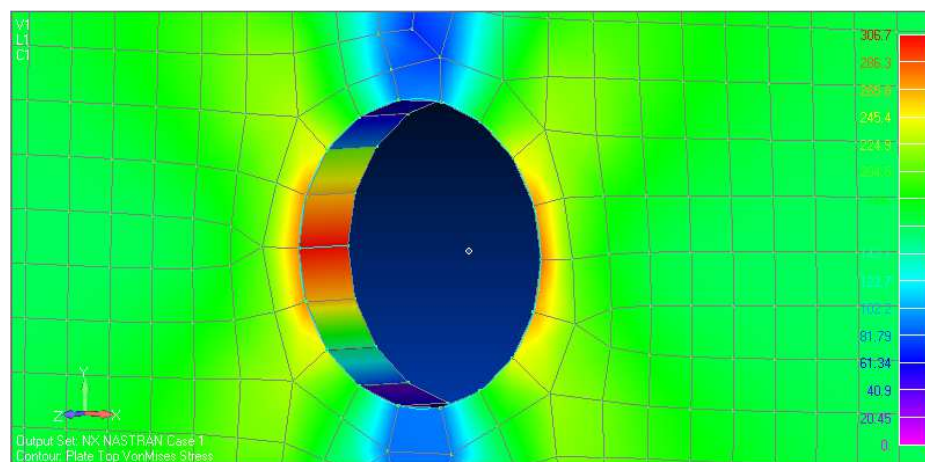


Figura 12 - Malha com elementos quadrados de uma camada

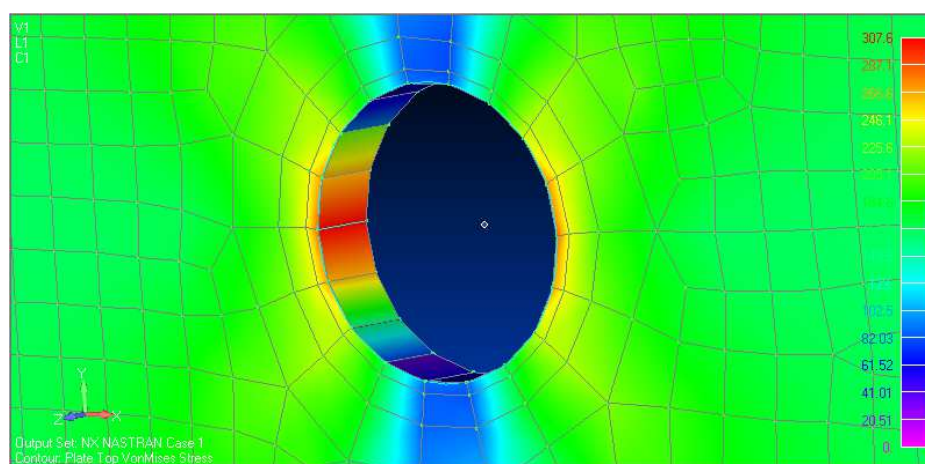


Figura 13 - Malha com elementos quadrados de três camadas

Agora serão mostradas imagens dos dados colocados sobre os elementos, para que se possa apreciar como a transição de tensões é mais radical no caso de uma camada do que no caso de três. O número de camadas possíveis pode ser selecionado no painel, no momento de malhar a peça, e pode variar de uma camada até um número de camadas determinado pelo usuário. Os valores das tensões na legenda estão expressos em MPa (FEMAP – Finite Element Modeling...).

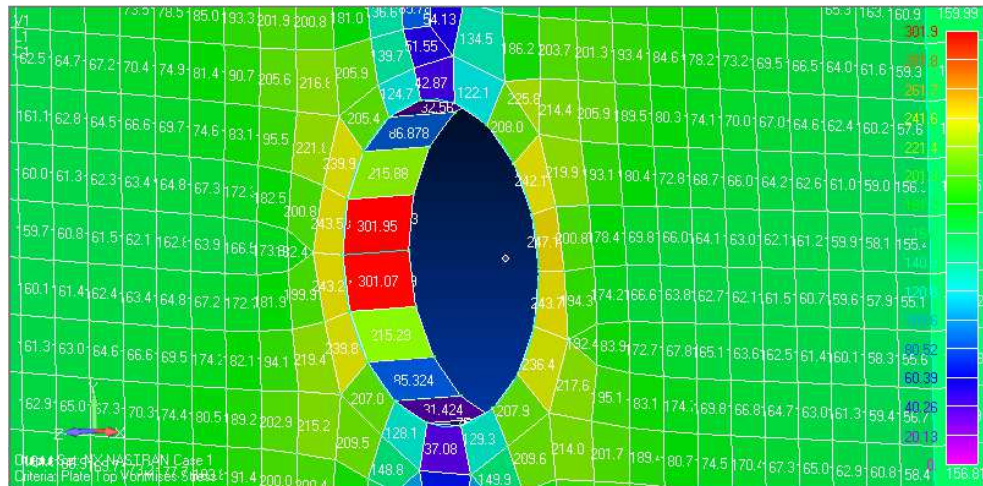


Figura 14 - Malha com elementos quadrados de uma camada com dados

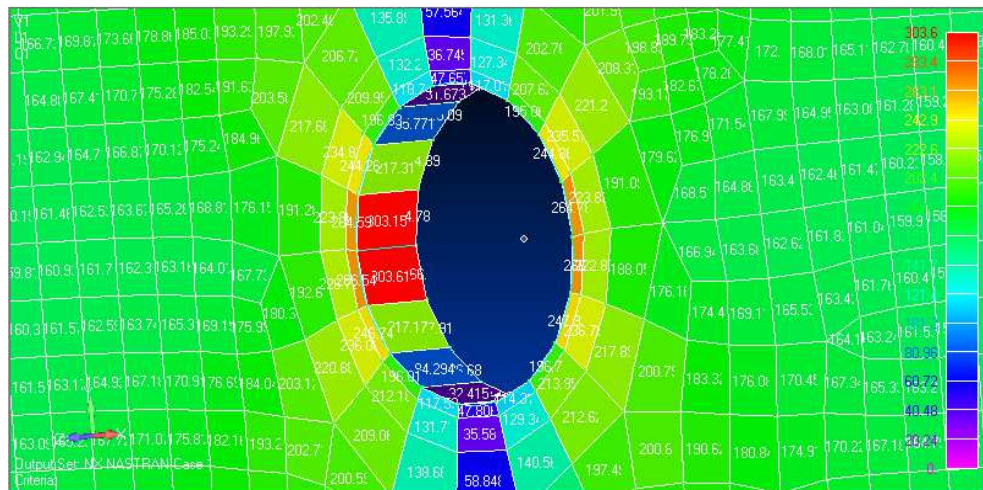


Figura 15 - Malha com elementos quadrados de três camadas com dados

3.2.5 Aplicação das condições de contorno

Neste ponto serão aplicadas condições de contorno (as cargas e as restrições) na peça para sua posterior análise.

- I. **Restrições:** Simula-se que a superfície superior da peça fica engastada, logo não se pode movimentar em nenhuma direção e, também, não pode girar. Como aparece mostrado na Figura 16.

- Caminho: *Model/Constrain/Nodal*

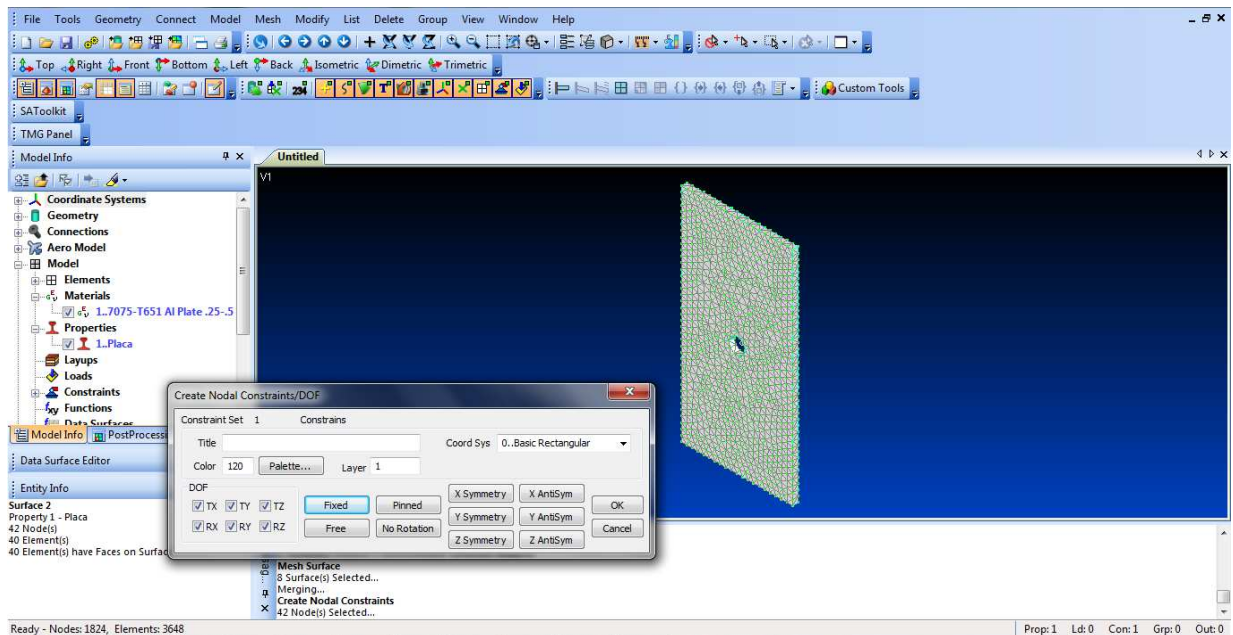


Figura 16 - Colocação das restrições

- II. *Cargas*: Uma carga no outro extremo da peça será aplicada de modo a tracionar a peça. Essa carga vai ser a maior carga que o material poderia suportar. Para poder analisar o comportamento da peça no seu ponto mais extremo, esta carga foi determinada devido ao conhecimento das propriedades do material, o que facilitou determinar a carga máxima que a peça pode suportar. A carga foi aplicada de forma distribuída sobre a superfície livre da peça.

O processo de seleção do carregamento utilizado no teste será explicado: Em primeiro lugar foi selecionado, de *data sheets*, a tensão de escoamento do material pode suportar, neste caso são 500MPa (*Aluminum 7075 Heat Treated (T6) Wrought data sheet*). Posteriormente pesquisou-se um coeficiente de segurança apropriado para o tipo de teste que se está realizando. De acordo com a FAR Sec.25.305 (*Federal Acquisition Regulation*) o fator de segurança mínimo aplicável para carregamentos deve ser 1,5. Conhecendo assim a tensão admissível (σ_{adm}) para o material (Equação(32)), o carregamento distribuído sobre superfície que não fica engastada será calculado (FEMAP – Finite Element Modeling...).

$$\sigma_{adm} = \frac{\sigma_e}{cs} = \frac{500MPa}{1,5} = 333,3MPa \quad (32)$$

Com a tensão admissível conhecida, um estudo das tensões máximas admissíveis no material está mostrado no **APÊNDICE A**.

Para dimensionar o carregamento, a tensão uniaxial T (Equação(33)) é calculada, e com essa tensão já conhecida, somente deve-se calcular a força (Equação(34)) e distribuí-la sobre a superfície para saber o carregamento que será colocado no software.

$$\sigma_{adm} = 3T \rightarrow T = 111,3MPa \quad (33)$$

$$T = \frac{F}{A} \rightarrow F = 111,3MPa \times (200 \cdot 10^{-3}m \times 10 \cdot 10^{-3}m) = 222,6kN \quad (34)$$

Já sendo conhecida a força (Equação (34)), esta será distribuída para saber qual valor será colocado no teste (Equação (35)).

$$q = \frac{F(N)}{d(m)} = \frac{222,6 \cdot 10^3 N}{200mm} = 1113 N / mm \quad (35)$$

Na Figura 17 observa-se como se coloca o carregamento na direção negativa do eixo y.

- Caminho: *Model/Load/Nodal*

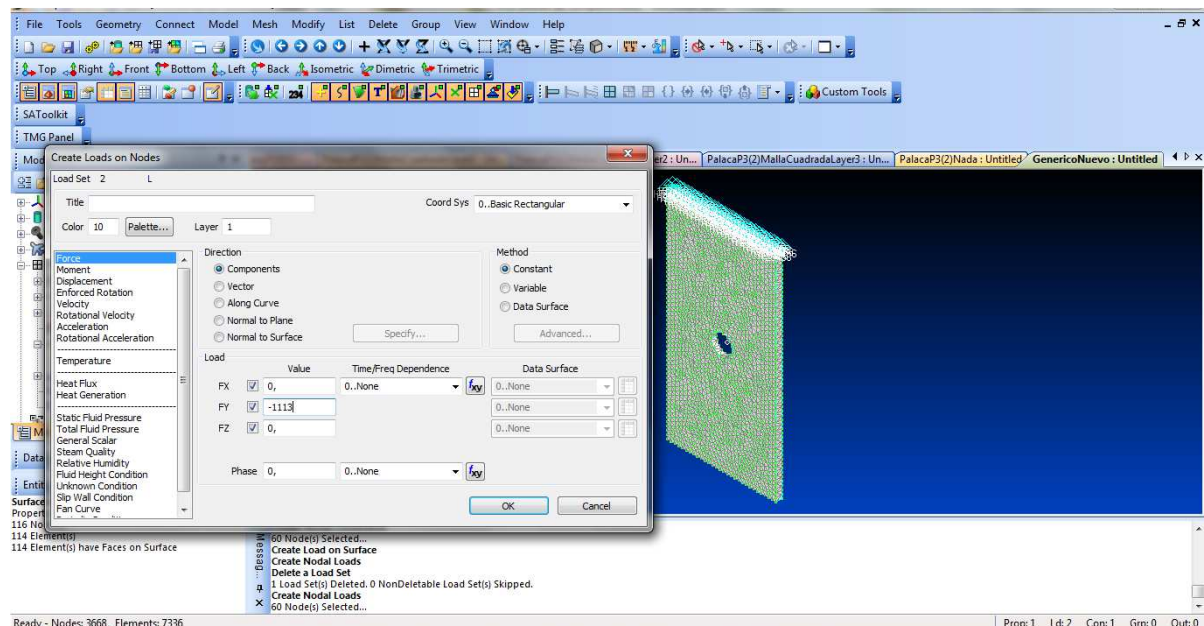


Figura 17 - Colocação do carregamento

No final a peça fica carregada e restrita como aparece na Figura 18.

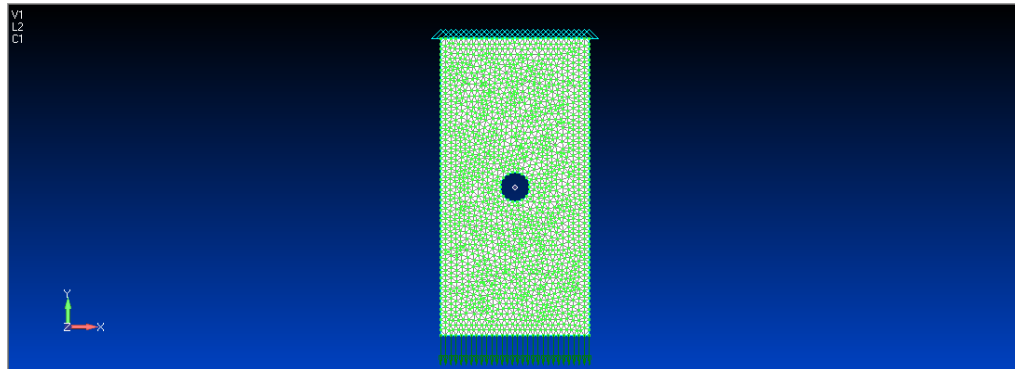


Figura 18 - Peça restrita e carregada

3.2.6 Análise da peça mediante NASTRAN®

Tendo a peça já carregada e restrita, com as propriedades do material já determinadas, apenas falta determinar que tipo de ensaio seja realizado. Um ensaio estático será simulado pelo sistema NASTRAN® (FEMAP – Finite Element Modeling...).

A Figura 19 mostra a análise da peça. Como pode se observar pela figura, as cargas e as restrições estão aplicadas e o ensaio determinado é do tipo estático, pedindo para o programa uma série de dados de saída.

- Caminho: *Model/Analysis/New/Analyze*

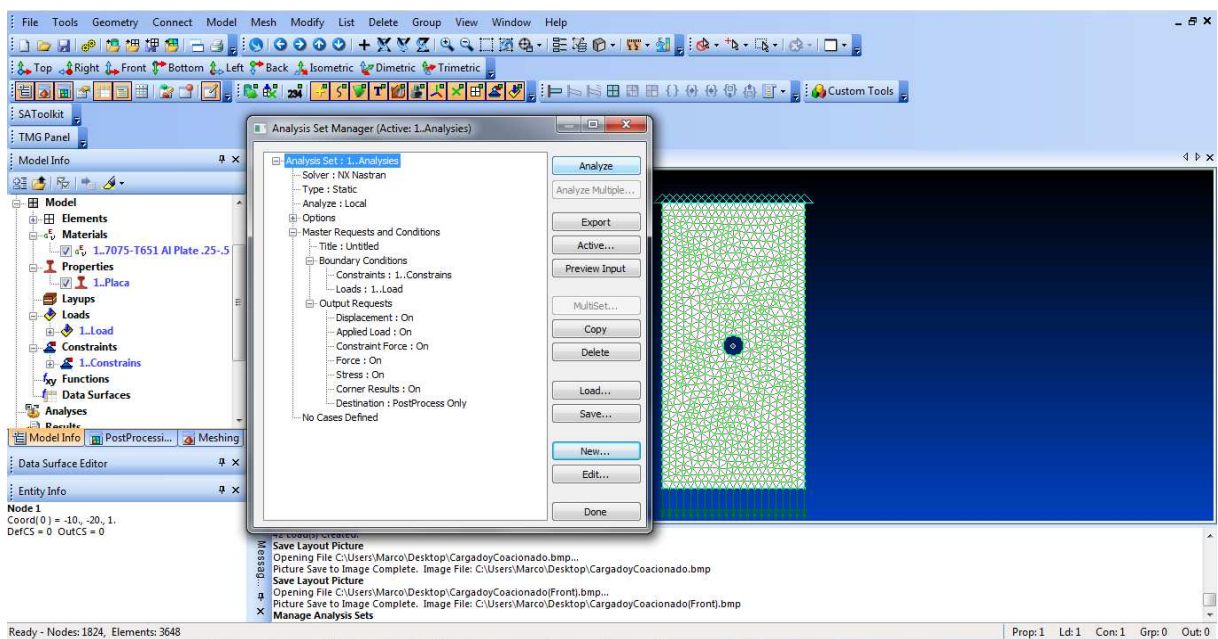
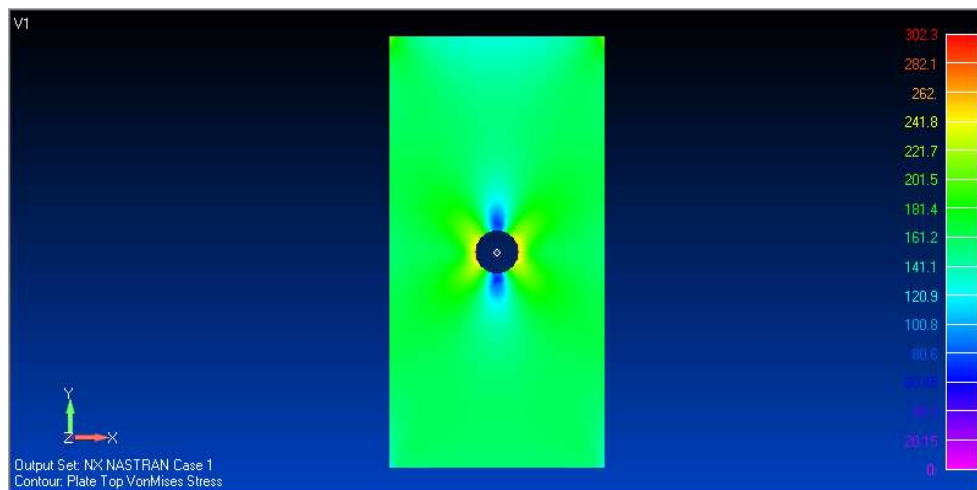


Figura 19 - Análise da peça

3.2.7 Pós-processamento dos dados

Após a análise realizada pelo programa NASTRAN[®] se tem a possibilidade de analisar os resultados obtidos da simulação. O programa aporta uma quantidade inumerável de opções, das quais serão selecionadas apenas as que interessam para o estudo. Algumas das opções utilizadas para o processamento dos resultados serão comentadas (FEMAP – Finite Element Modeling...).

- I. Em primeiro lugar, a distribuição das tensões sobre a peça pode ser observada, onde os distintos valores se diferenciam através das cores (Figura 20), estes valores vêm determinados por uma legenda de cores. Os valores são dados em MPa.

*Figura 20 - Peça analisada mediante as tensões de VonMisses*

Este teste em particular foi realizado pelo critério 7033 - *VonMisses Stress*, mas o software permite analisar os diferentes estados tensionais da peça e proporciona uma larga lista de possíveis testes para sua análise, como pode ser observado na Figura 21.

- Caminho: *PostProcessing/Post data*

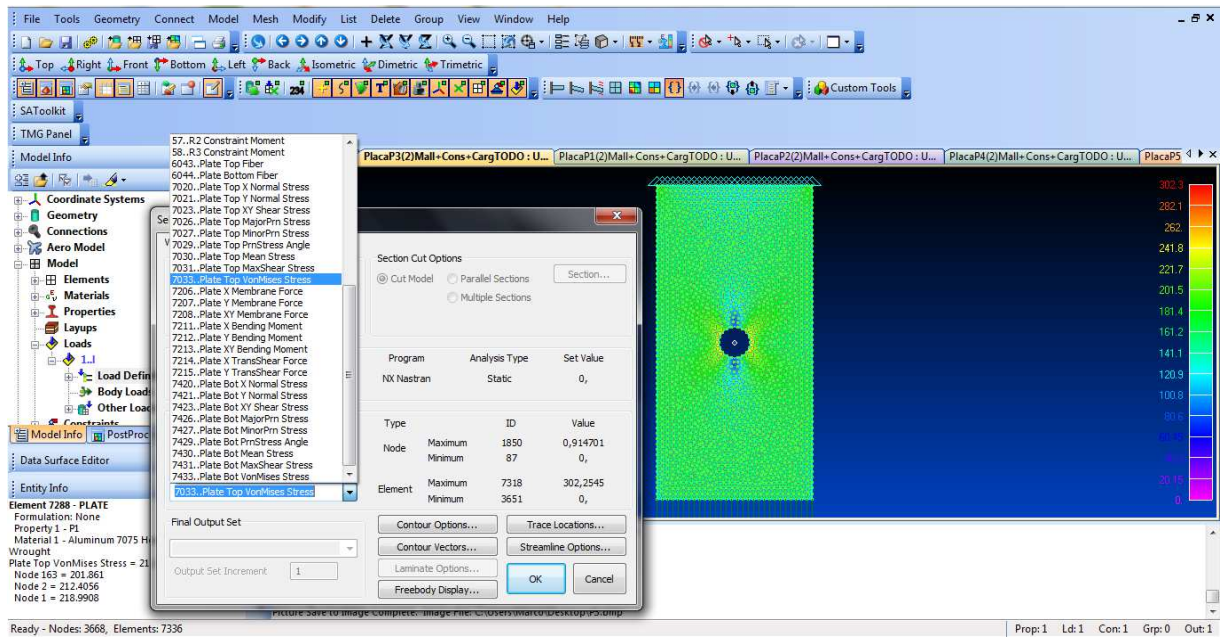


Figura 21 - Diferentes análises possíveis

Dependendo da necessidade, pode se escolher entre as diferentes opções de ensaios que o software oferece. Em particular, neste trabalho a maior parte dos ensaios será realizada com o critério das tensões de *Von Mises*. Para isto será recordado o critério de seleção no **APENDICE B**.

- II. Continuando com o pós-processamento dos dados, o software proporciona tanto animações de como a peça é deformada até imagens de como ficará depois de ter suportado o carregamento.

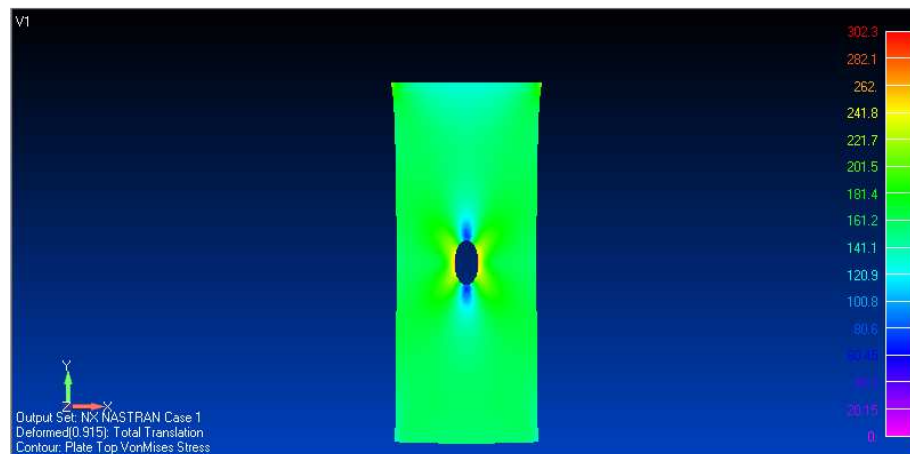


Figura 22 - Peça deformada

Na Figura 22 pode se observar como a peça fica deformada, e o furo central se torna uma elipse.

- III. O software também proporciona ferramentas para apreciar os diferentes ângulos de vista da peça mediante um sistema de visão de 360°. Além disso, pode-se cortar as peças por diferentes planos para observar como se distribuem as tensões no interior deles (Figura 23).

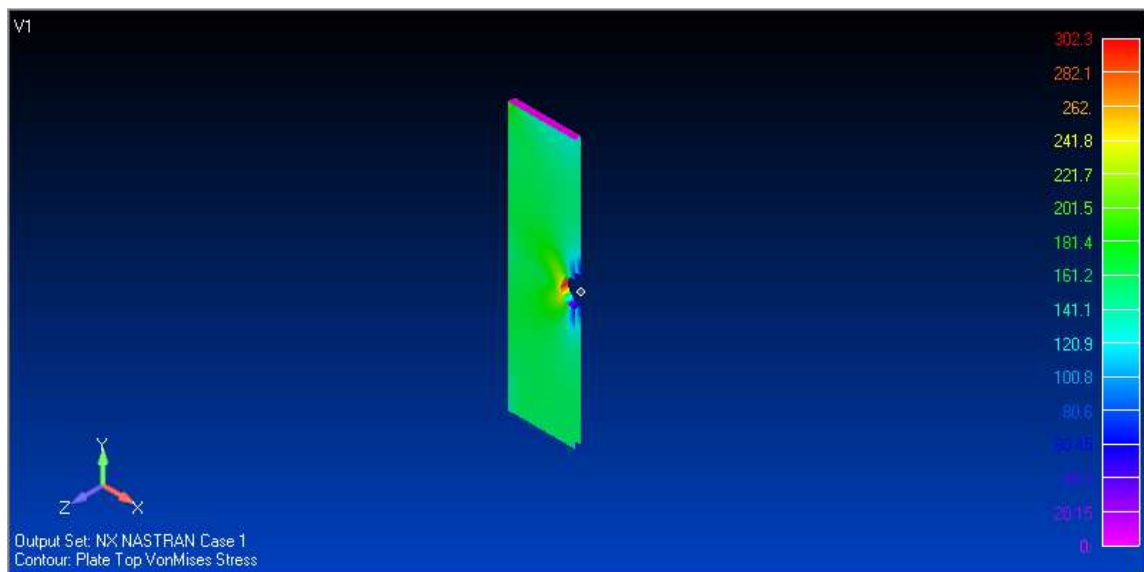


Figura 23 - Peça analisada e cortada, para ver as tensões no furo

Nesta Figura 23 pode-se observar como as tensões máximas ficam localizadas nos extremos laterais do furo central da peça, e como se ajusta com a realidade, já que a teoria descreve que devem estar localizados nesses pontos.

Na Figura 24 foi realizado um zoom na zona de importância da peça para poder observar como maior precisão as zonas desejadas (FEMAP – Finite Element Modeling...).

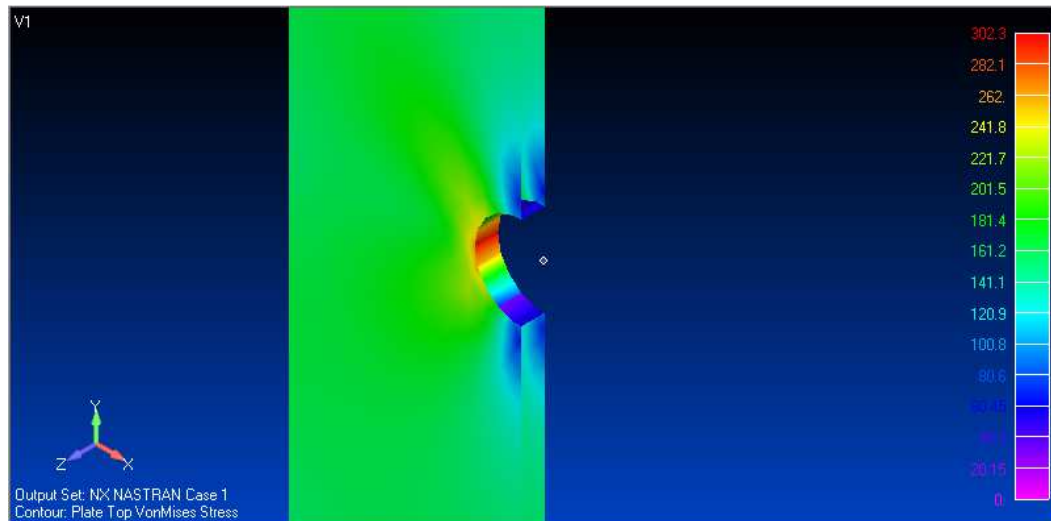


Figura 24 - Zoom da peça cortada no furo

Estas são algumas das ferramentas que o programa FEMAP[®] pode proporcionar para as análises de estruturas mediante os elementos finitos.

3.2.8 Tomada de dados

Após ter sido feita a análise da peça e conhecer algumas das opções que o software pode oferecer, foi preciso fazer análises com diferentes furos na peça e em distintas posições na mesma. Também, foram selecionados dois diferentes tamanhos de furo (de raios 15mm e 20mm) para poder obter os dados, que posteriormente serão utilizados no *software* MATLAB[®], no código das redes neurais.

Com as peças já analisadas, quatro informações para cada peça foram obtidas (cada informação representando um sensor). As informações foram obtidas nas posições X1, X2, X3 e X4 (Figura 25). A posição da retirada de dados foi a mesma para cada peça independente da localização do furo e de seu tamanho. Estas posições ficam próximas dos cantos da placa, e podem ser mais bem apreciadas na Figura 25.

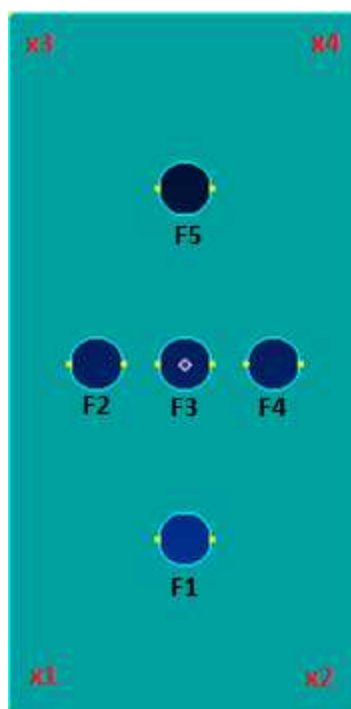


Figura 25 - Localização dos furos e dos sensores

Nesta Figura 25 pode-se perceber onde foram realizados os furos (somente um furo por placa foi considerado por vez) dentro de cada placa, e onde foram tomados os valores dos dados, que estão representadas pelas cruzes vermelhas (X1, X2, X3 e X4) que estão localizadas nos cantos da peça. Então, pode-se elaborar duas tabelas (Tabela 1 e Tabela 2) com todos os dados dos diferentes pontos, posições e tamanhos dos furos. Os valores de X1, X2, X3 e X4 são valores de tensão.

Tabela 1 - Dados tomados dos testes das peças de R=15mm.

R=15	X1(MPa)	X2(MPa)	X3(MPa)	X4(MPa)	MAX(MPa)
F1	160,7	160,67	165,1	163,85	270
F2	160,54	160,55	159,87	163,88	268
F3	156,44	156,36	164,88	164,74	286,6
F4	157,34	156,25	163,22	158,96	270
F5	158,31	156,33	169,89	169,47	265,7

Tabela 2 - Dados tomados dos testes das peças de R=20mm.

R=20	X1(MPa)	X2(MPa)	X3(MPa)	X4(MPa)	MAX(MPa)
F1	157,04	157,16	160,87	160,86	308,4
F2	156,42	156,68	163,54	163,98	317,5
F3	157,74	156,65	165,42	159,74	302,3
F4	157,39	156,35	163,32	164,06	318
F5	157,34	156,25	163,22	158,96	302,6

Todos os valores das tabelas estão dados em MPa, e como pode-se observar, são colocados valores tomados nos diferentes pontos da peça, começando desde o canto inferior esquerdo que corresponde ao valor de X1, o canto inferior direito que corresponde ao valor de X2, e respectivamente para os cantos superiores com os valores de X3 e X4.

Na última coluna foram colocados os valores máximos que se localizam em cada um dos testes, logicamente estes valores estão localizados no furo, e como se pode apreciar, estes valores são aproximadamente três vezes o valor da tensão colocada no extremo inferior (111,3MPa).

O software FEMAP[®] tem a possibilidade de mostrar em imagens os diferentes valores numéricos das análises escolhidas. Na Figura 26 pode se apreciar mais de perto como ficam os valores no centro de cada elemento, além de ter distinções por cores (FEMAP – Finite Element Modeling...).

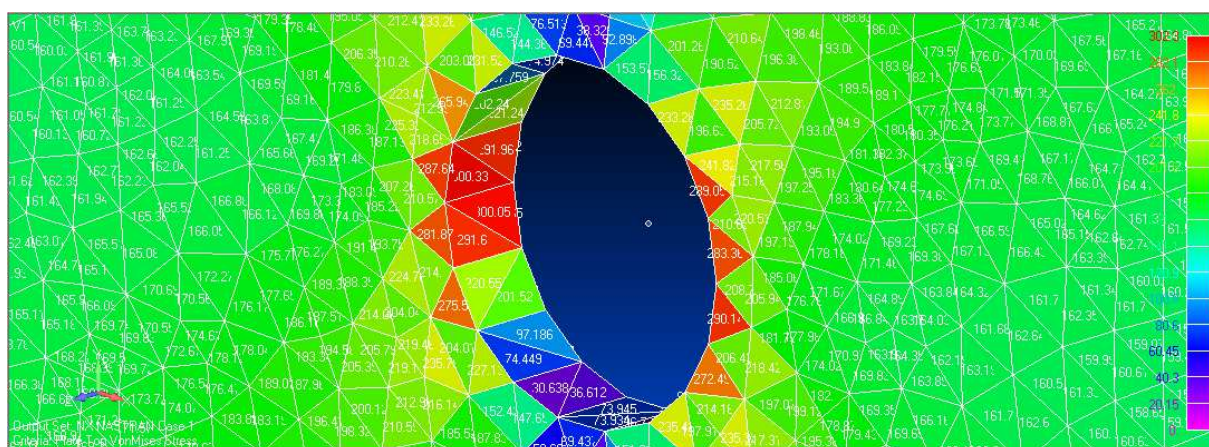


Figura 26 - Zoom do furo com os dados

CAPÍTULO 4 - PRINCÍPIOS FUNDAMENTAIS DAS REDES NEURAIIS ARTIFICIAIS

4.1 Introdução

Um dos principais fatores que diferenciam o ser humano dos animais é a capacidade de ser racional, e a partir desta capacidade pode-se desenvolver a tecnologia existente na atualidade. Por isso, há um grande interesse em criar modelos artificiais que se assemelham ao funcionamento do cérebro humano. Este campo é denominado inteligência artificial (AI – *Artificial Intelligence*), onde se localizam as redes neurais artificiais que interessam para a realização deste trabalho.

4.2 Revisão Histórica

Os primeiros teóricos que conceberam a computação neuronal foram Warren McCulloch, um neurofisiólogo, e Walter Pitts, um matemático, em 1943. Eles lançaram uma teoria sobre o funcionamento dos neurônios no livro (“Um Cálculo Lógico da Iminente Idéia da Atividade Nervosa” - Boletim de Matemática Biofísica 5: 115 – 133). Posteriormente Donald Hebb em 1949, no livro que escreveu explicou os processos de aprendizagem de um ponto de vista psicológico, e desenvolveu uma regra de como ele ocorria. Ainda hoje o fundamento dos métodos de aprendizagem segue a mesma regra. Os trabalhos de Hebb são as bases da Teoria das Redes Neurais. Uma data importante dentro da história das redes neurais foi em 1956, quando aconteceu o Congresso de Dartmouth onde se determinou o nascimento da Inteligência Artificial.

Um ano depois Frank Rosenblatt (em 1957) começou a desenvolver o *perceptron*, que é a rede neural mais antiga. Este modelo era capaz de reconhecer padrões após realizar um processo prévio de aprendizagem. As investigações de Minsky e Papert (1959) frearam o desenvolvimento das redes neurais já que publicaram um livro onde se provava matematicamente que o *perceptron* era fraco porque não era capaz de resolver problemas não lineais, que são muito utilizados na computação. Então, em 1985, com a publicação do livro “Computação neuronal de decisões em problemas de otimização”, por John Hopfield que

ressurgiu o interesse pelas redes neurais. Um ano depois, David Rumelhart e G. Hinton (em 1986) redescobriram o algoritmo de aprendizagem de retro-propagação (propagação para trás) proposto por Paul Werbos em 1974. Em 1986 as redes neurais tomaram maior importância e iniciaram-se os trabalhos e estudos a seu respeito.

4.3 Conceito de rede neuronal e vantagens

Pode se definir um neurônio como a menor unidade dentro da estrutura do cérebro. Fazendo uma comparação do cérebro com um computador, os neurônios poderiam ser comparados com as unidades de silício dos computadores, estes têm aproximadamente cinco vezes a velocidade dos neurônios, mas o cérebro supre esta deficiência com um número maior de interconexões e com a capacidade de operar em paralelo, e resolver as situações em um tempo menor que os computadores.

Não existe uma definição exata de neurônio artificial, cada fonte consultada terá uma definição, mas todas elas tem uma base comum, que consiste em tentar simular o comportamento do cérebro. Algumas das definições, por exemplo, são:

- Modelos matemáticos desenvolvidos para imitar o cérebro humano (CHEN, 1998);
- Modelo computacional, em paralelo, formado por unidades processadoras adaptativas com uma alta interconexão entre elas (LIN, 1996).

Os elementos neurais estão interconectados entre si mediante os chamados pesos sinápticos. Estes pesos sinápticos variam de acordo com o tempo, já que variam conforme o processo de aprendizagem. Este processo de aprendizagem pode ser definido como a modificação das conexões entre os neurônios, os pesos sinápticos, para realizar uma tarefa determinada (SERRANO, 2009).

As principais vantagens das redes neurais são:

- *São sistemas que podem tolerar falhas:* O sistema pode tolerar falhas dentro dele e de alguns dos neurônios sem alterar de um modo substancial a resposta do sistema;
- *Grande adaptabilidade:* Uma rede neuronal é capaz de modificar parâmetros de seu funcionamento perante mudanças que aconteçam no processo do trabalho;
- *São sistemas não lineares:* Um neurônio é um elemento não linear, portanto uma rede de neurônios também será um sistema não linear.

4.4 Modelos neurais

Neste ponto, será explicada a teoria de um modelo neural e o seu funcionamento. Num modelo artificial de neurônio, além de existirem as entradas, que são os dados que se conhecem, e as saídas, que são os dados a serem conhecidos, há três elementos principais:

- Conjunto de conexões e pesos, que determinam o comportamento do neurônio. Estas conexões podem ser excitadas, onde os valores dos pesos serão positivos e terão uma tendência a incrementar o nível de atividade dos neurônios. Ou poderão ser inibidoras, onde os pesos serão negativos e terão uma tendência a reduzir o sinal de saída do neurônio;
- Um somador, que será o encarregado de somar todas as entradas e multiplicá-las pelos pesos de todas as sinapses;
- Uma função de ativação, normalmente não linear, que se aplica à saída do somador para limitar a amplitude de saída do neurônio. Muitas funções de ativação tem sido utilizadas em redes neurais:
 - Funções lineares, que somente multiplicam as entradas por uma constante para obter as saídas.
 - Função *signum*, que foi a primeira função apresentada por McCulloch e Pitts (1987) e apresenta uma separação entre os dados e está definida na Equação (36)

$$F(n) \begin{cases} -1 & \text{se } n < 0 \\ 1 & \text{se } n > 0 \end{cases} \quad (36)$$

- Funções degrau, que deixa o valor de saída do neurônio igual a um, se o valor de entrada ultrapassa um determinado valor *threshold*, e igual a zero, se não. Como definida na Equação (37)

$$F(n) \begin{cases} 0 & \text{se } n < 0 \\ 1 & \text{se } n > 0 \end{cases} \quad (37)$$

- Função logística ou sigmoideal (Equação (38)) que tem duas características interessantes para ser usada como função de ativação. A primeira delas é que é muito não linear, a segunda que é contínua e diferenciável. Alguns algoritmos de aprendizagem precisam que a função de ativação seja diferenciável.

Características que a função *signum* e degrau não possuem. A equação da função pode ser vista na Figura 27.

$$F(n) = \frac{1}{1 + e^{-n}} \quad (38)$$

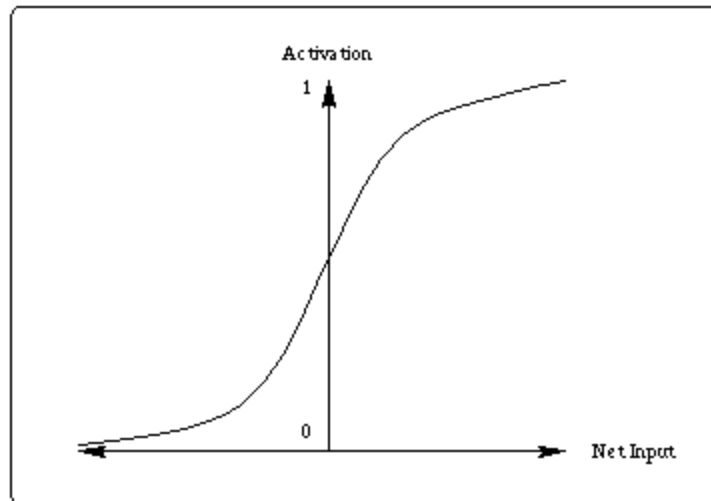


Figura 27 - Função sigmoidal

Depois de ter conhecido os principais elementos de uma rede neuronal, pode-se apreciar na Figura 28 como está determinada a estrutura de uma rede simples (SERRANO, 2009; IZAURIETA, SAAVEDRA; LOPES, 2007).

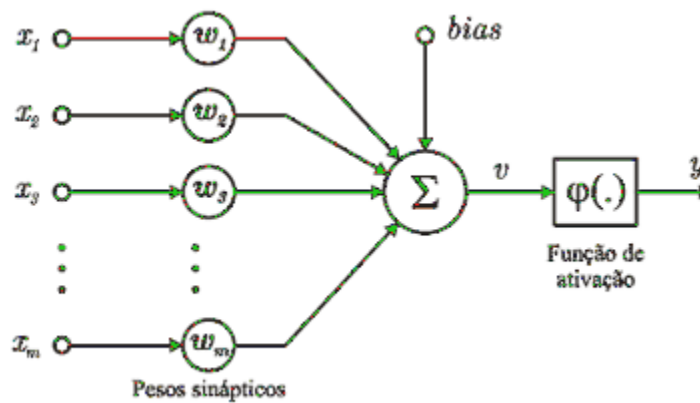


Figura 28 - Esquema modelo neural

Observando a Figura 28, os valores de x_i são interpretados como os sinais de entrada e os valores w_i como os pesos sinápticos nos dendritos. O que cada peso sináptico faz é multiplicar o valor da entrada pelo valor do peso, definindo a importância relativa dos neurônios. No somatório é feita a soma total das entradas multiplicadas pelos pesos sinápticos com se mostra na Equação (39). A entrada *bias* (b_i) no somatório tem o papel de aumentar ou diminuir a influência do valor das entradas.

$$y_j = \sum_{i=1}^n w_{ji} \cdot x_i + b_i \quad (39)$$

Os neurônios serão ativados quando a entrada supera um valor *threshold*, e para isto o que se faz é aplicar a função de ativação já definida anteriormente, que podem ser de diferentes tipos. Quando se quer aumentar ou reduzir este valor *threshold*, o que se faz é dar um valor a uma entrada x_0 ao que se atribui um valor de 1 (entrada *bias*) e um peso sináptico de $w_{i0} = b_i$ e ao somatório atribui-se um valor fixo de 0 (ou seja, o contador i começa em 0). Assim pode-se observar que o valor atribuído ao neurônio somador terá que ser o valor do peso sináptico, como pode se apreciar na Equação (40)

$$y_j = \sum_{i=0}^{Xn} w_{ji} \cdot x_i \quad (40)$$

Tendo estes conceitos claros, diferentes modelos de redes neurais podem ser realizados, dependendo do tipo as entradas, do valor *threshold* e das funções de ativação. Funções simples podem ser realizadas, como, por exemplo, a demonstração das funções lógicas mediante uma combinação apropriada de neurônios. Dessa maneira, pode-se demonstrar, utilizando como fizeram McCulloch e Pitts (1987), neurônios de tipo binário (0,1) com os valores dos pesos sinápticos constantes, igual ao valor *threshold* e uma função de ativação tipo degrau.

Além de poder resolver problemas simples, como os problemas de resolução das funções lógicas, as redes neurais são capazes de resolver problemas mais complexos, mediante a adição de camada, tomando a camada zero como a camada das entradas, e a última camada, a de saída. Costuma-se não contabilizar a camada das entradas. Portanto uma rede com a camada das entradas e a camada das saídas, como as redes de resolução dos problemas lógicos, serão denominadas redes unicamada.

Quando se trabalha com um grande número de neurônios é normal agrupar os que têm comportamentos similares em camadas, cada camada é um vetor de neurônios.

Numa rede neural com mais de duas camadas, as camadas ocultas, assim são chamadas as camadas intermédias, sempre são não lineares. Isso porque se fosse construída uma rede neural com camadas ocultas lineares, esta seria equivalente a uma rede unicamada. O comportamento dos neurônios dentro de uma mesma camada é em paralelo, já que cada neurônio trabalha de maneira independente e simultânea com os outros neurônios (SERRANO, 2009; IZAURIETA, SAAVEDRA; LOPES, 2007).

Na Figura 29 pode se apreciar as diferentes camadas de uma rede neural.

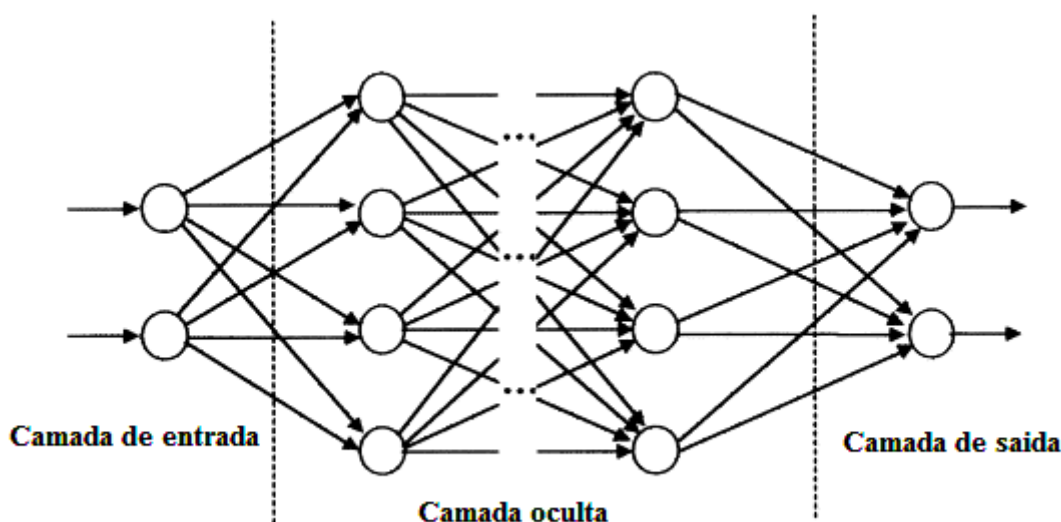


Figura 29 - Esquema das camadas da rede neuronal

4.5 Processo de aprendizagem

O processo de aprendizagem é um ponto chave dentro das redes neurais, já que nele se adaptam as sinapses dos neurônios para que a rede responda de um jeito diferente aos estímulos do meio. Toda informação relativa ao processo é guardada nos pesos sinápticos dos neurônios. O que diferencia os seres humanos dos animais é o número de conexões sinápticas, a organização e a troca de informação entre elas, isto porque os tipos de neurônios são iguais para todos.

Os processos de aprendizagem poderão ser divididos em dois tipos, supervisionado, que é o tipo de aprendizagem que se utilizará neste trabalho, e o não supervisionado. Será dada maior ênfase no processo de aprendizagem supervisionado.

O processo consiste em ‘ensinar’ à rede quais são as respostas corretas às perguntas cujas soluções já são conhecidas. O processo de aprendizagem será explicado utilizando uma analogia. Supondo-se que se quer ensinar a uma criança as cores, os passos serão os seguintes (SERRANO, 2009; IZAURIETA, SAAVEDRA).

1. Tem-se um conjunto diferente de cores desenhado num papel que irá ser mostrado para a criança, tendo em mãos a resposta correta.
2. Será introduzida uma entrada e espera-se a resposta da rede. É a mesma coisa que mostrar à criança a cor preta e perguntar para ela “*qual cor é esta?*”.
3. A rede neural dará uma resposta. O que será feito é comparar a resposta dada pela rede com a resposta certa conhecida. Será gerado um vetor de erro que se utilizará para corrigir a sinapse neuronal mediante um algoritmo. Ao se obter a resposta da criança “*a cor é verde*”, então se fala para ela “*essa cor não é verde, é a cor preta*”.
4. A sequência de aprendizagem dos r pares de treinamento é dado o nome de época. Terá tantas épocas quanto se precise para que o sistema forneça a resposta correta. Para adaptar as sinapses se utilizará a Equação (41). Onde os valores dos pesos sinápticos são $w_{ji}(n)$, o valor de $\varepsilon w_{ji}(n)$ é uma pequena alteração no valor dos pesos com respeito a seu estado anterior. O que diferenciam os processos de aprendizagem e os algoritmos é principalmente como achar este $\varepsilon w_{ji}(n)$.

$$w_{ji}(n+1) = w_{ji}(n) + \varepsilon w_{ji}(n) \quad (41)$$

Dentro deste tipo de aprendizagem supervisionado encontramos dois algoritmos fundamentais, que são:

- Perceptron: Um *perceptron* é uma rede de uma só camada, onde os neurônios de saída são não lineares e tem funções de ativação do tipo degrau. Este algoritmo não é útil para o caso em estudo neste trabalho já que não resolvem problemas complexos, logo, vai-se concentrar no problema das redes multicamada.
- Backpropagation: É o algoritmo usado em redes multicamada. Este algoritmo é um algoritmo iterativo já que não existe uma solução analítica iterativa, e permite treinar as redes multicamada minimizando o erro cometido pela rede. Pode-se definir o erro cometido como a

soma dos quadrados dos erros individuais. Cada padrão de treinamento se propaga pela rede para produzir uma resposta na camada da saída, onde é comparado com os padrões objetivos. Com estes valores testados e os objetivos, pode-se obter o erro de aprendizagem, este erro será o que marca o caminho para atualizar os pesos e obter a saída desejada. Isso é conseguido, minimizando-se o erro quadrático médio em cada iteração (SERRANO, 2009; IZAURIETA, SAAVEDRA; LOPES, 2007). O fundamento básico da rede é o seguinte:

- Calcular o erro em cada saída para cada padrão;
- Ajustar os pesos sinápticos na camada da saída para diminuir o erro;
- Propagar os erros até a camada da entrada ajustando os pesos das camadas ocultas;
- Repetir este processo de forma iterativa.

O processo visto de uma forma matemática é o seguinte:

1. Tem-se que definir um padrão de treinamento com os componentes da Equação (42)

$$p = \begin{pmatrix} p_1 \\ p_2 \\ \cdot \\ \cdot \\ p_q \end{pmatrix} \quad (42)$$

Sendo:

q - número de componentes do vetor de entrada;

m - número de neurônios da camada oculta;

l - número de componentes da camada de saída.

2. Este padrão de treinamento gerará uma entrada n em cada uma dos neurônios da camada seguinte, como na Equação (43). O valor zero representa a camada a que cada valor pertence.

$$n_j^0 = \sum_{i=1}^q (w_{ij}^0 \cdot p_i) \quad (43)$$

Sendo:

w_{ij}^0 - peso que une a componente i da entrada com o neurônio j da camada oculta;

p_i - componente i do vetor p de treinamento.

Cada neurônio da camada oculta tem como saída a_j^0 (Equação (44))

$$a_j^0 = f^0 \left(\sum_{i=1}^q w_{ji}^0 \cdot p_i \right) \quad (44)$$

Sendo:

f^0 - função de transferência dos neurônios da camada oculta.

As saídas a_j^0 dos neurônios da camada oculta (que tem m componentes) são as entradas aos pesos de conexão da camada de saída, que está descrito por n_k^s , na Equação (45)

$$n_k^s = \left(\sum_{j=1}^m w_{kj}^s \cdot a_j^0 \right) \quad (45)$$

Sendo:

n_k^s - entrada ao neurônio k da camada de saída;

w_{kj}^s - peso que une ao neurônio j da camada oculta com o neurônio k da camada de saída (que tem s neurônios);

a_j^0 - saída do neurônio j da camada oculta (que tem m neurônios).

A rede produz uma saída final a_k^s como na Equação (46)

$$a_k^s = f^s(n_k^s) \quad (46)$$

Sendo:

f^s - função de transferência dos neurônios da camada de saída.

3. A saída a_k^s se compara com a saída desejada t_k para calcular o erro em cada unidade. Como na Equação (47).

$$\delta_k = (t_k - a_k^s) \quad (47)$$

4. O erro devido a cada padrão p propagado está dado pela Equação (48)

$$ep^2 = 1/2 \sum_{k=1}^s (\delta_k^2) \quad (48)$$

Sendo:

ep^2 - erro quadrático médio para cada padrão de entrada;

δ_k - erro no neurônio k da camada de saída com l neurônio.

5. Este processo se repete para o número total de padrões de treinamento r . O objetivo do algoritmo é atualizar todos os pesos minimizando o erro quadrático médio descrito na Equação (49).

$$e^2 = \sum_{p=1}^r (ep^2) \quad (49)$$

Sendo:

e^2 - erro total no processo de aprendizagem em uma iteração.

6. O erro que gera uma rede neural estará determinado por seus pesos específicos, e gerará um espaço dimensional em função do número de conexões da rede. Ao se avaliar o gradiente do erro neste espaço se obterá uma direção na qual o erro terá um maior crescimento, e o objetivo do processo de aprendizagem é minimizar o erro. Terá que se tomar a direção negativa do gradiente para obter o maior decrescimento do erro, e assim sua minimização (SERRANO, 2009; IZAURIETA, SAAVEDRA).

CAPÍTULO 5 - ESTUDO DO PROGRAMA MATLAB

5.1 História do MATLAB

MATLAB[®] (*MATrix LABoratory*) foi criado em 1984 por Clever Moler com a idéia de implementar pacotes de sub-rotinas escritas em Fortran em um curso de álgebra, para não ter que escrever os programas nessa linguagem. Hoje em dia pertence à empresa MathWorks[®].

MATLAB[®] é um *software* matemático que tem um ambiente de desenvolvimento integrado com uma linguagem de programação própria (a linguagem M). Com ele pode se programar algoritmos, representar funções e dados, criar interfaces de usuário e comunicar-se com outros programas em distintas linguagens de programação.

Este software é muito utilizado dentro de Universidades e centros de ensino, e também é utilizado no âmbito industrial para a investigação e para resolver problemas práticos da engenharia. Pode-se falar que tem praticamente um milhão de usuários no mundo (Soporte MATLAB[®]).

5.2 Processo de utilização do software MATLAB[®] para implementação das redes neurais.

O programa MATLAB[®] é em si mesmo uma forte ferramenta de cálculo e já tem algumas funções e códigos já implementados para sua utilização. Um destes códigos é o das redes neurais.

O que vai ser feito aqui é aproveitar o código de redes neurais para introduzir os dados de tensões tirados das análises das peças no programa FEMAP/NASTRAN[®]. Além disso, após introduzi-los no código das redes neurais, depois os resultados obtidos serão analisados. O processo será o seguinte:

- Criar um vetor de entrada [P] com os valores das tensões tiradas das peças;
- Criar um vetor de saída [T] com as posições de todos os furos;
- Criar a rede neural chamada 'net' mediante o comando:

```
net=newff(P,T,[N1 N2],{'tansig' 'tansig' 'purelin'},'traingdx')
```

Sendo:

- P é o vetor de entradas;
 - T é o vetor de saída;
 - N1 é o numero de neurônios da primeira camada oculta;
 - N2 é o numero de neurônios da segunda camada oculta;
 - Esta rede terá funções de transferência tangente sigmoidal entre as camadas de entrada e a primeira camada oculta, e entre as duas camadas ocultas. Haverá uma função linear entre a segunda (ou última) camada oculta e a saída;
 - O último parâmetro “traingdx” vai determinar o tipo de treinamento ao que a rede será submetida. Neste caso será uma aprendizagem supervisionada adaptativa de acordo com o algoritmo *backpropagation*, junto com um gradiente descendente.
- Treinamento da rede neural por meio da função “train”, onde os valores dos pesos sinápticos serão ajustados mediante o método “traingdx”. Já que se conhecem as entradas e saídas, o algoritmo ajustará os pesos até que se obtenham os valores das saídas. Utilizar-se-á o seguinte comando pra isso:

```
net=train(net,P,T)
```

- Já com os pesos sinápticos ajustados, falta somente simular a rede para saber se a rede criada e treinada é capaz de resolver o problema. É importante deixar o valor da tensão e a posição do furo que vai ser testado fora dos vetores [P] e [T] para poder saber se a rede é capaz de achá-lo. Utilizar-se-á o comando ‘sim’ que dará como resultado o valor do raio e a posição x e y do possível furo.

```
sim(net,P)
```

Em um pequeno resumo, o que se vai fazer é criar uma rede neural no programa com as camadas desejadas, e treiná-la com nove pares de tensão-posição (entrada-saída), será deixado um para ver a resposta da rede.

Na Figura 30 pode se apreciar a janela que se abre quando está se analisando o código. Onde pode se apreciar na parte superior o esquema do processo com as oito entradas, a primeira camada oculta formada por cinquenta neurônios, e uma função de transferência

sigmoidal, a segunda camada formada por 20 neurônios, e também uma função de transferência sigmoidal. A camada de saída com uma função linear e três neurônios correspondentes ao raio, a posição x e a posição y .

Pode se apreciar também na Figura 30 o tipo de treinamento que o método “traingdx” está submetido e o número de épocas, que corresponde a 5000. Também aparece o tempo utilizado para fazer a análise, e a execução do processo com gradiente.

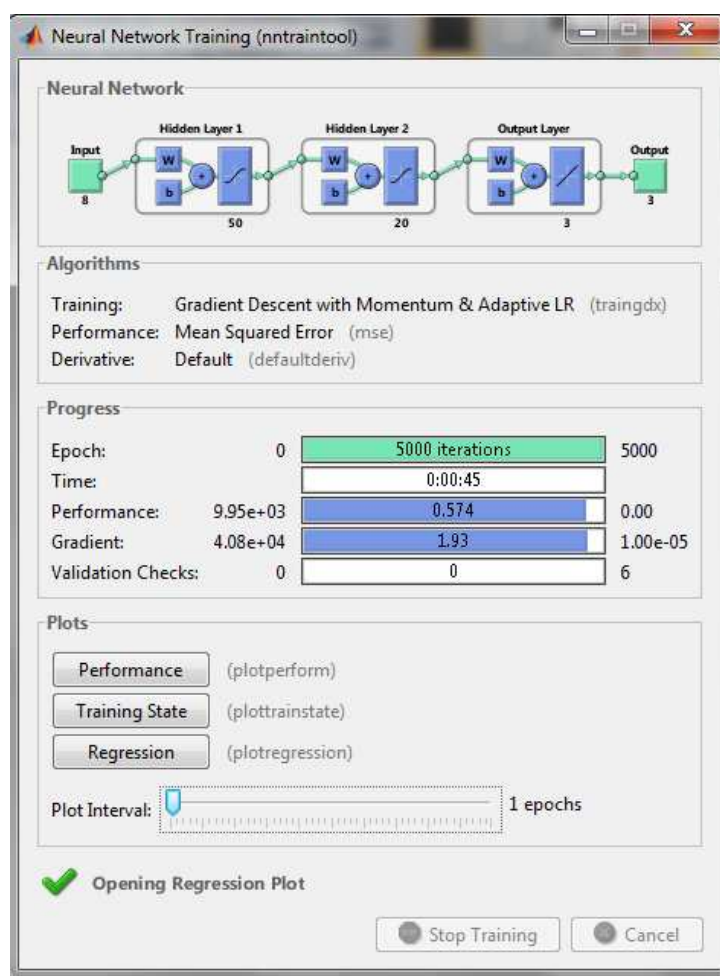


Figura 30 - Esquema da janela do processo

Deve-se falar que o processo de ajuste das redes neurais é um processo que não tem um padrão determinado. O ajuste dos elementos que a compõe depende de cada usuário. Neste trabalho determinou-se fazer uma comparação deixando fixa a primeira camada oculta com cinquenta neurônios, e variando a segunda camada oculta entre dez e vinte, mantendo um número de épocas de cinco mil.

Observa-se abaixo o código utilizado para a realização da simulação, particularizado para o cálculo da posição do furo número quatro, de coordenadas (50,0) cm.

Código utilizado no programa

```
T=[15 0 -100;15 -50 0;15 0 0;15 0 100;20 0 -100;20 -50 0;20 0 0;20 50 0;20
0 100]%Vector de posicoes(saida)

P=[162.85 146.14 162.75 160.65 158.53 158.32 148.35 160.07;159.7 160.13
160.9 157.37 158.01 149.02 158.58 168.04;159.89 159.58 159.33
159.02 158.52 145.45 158.44 159.25;158.15 159.72 160.99 160.37
163.18 130.33 161.13 159.85;165.75 138.56 167.19 162.58 157.22
143.66 158.63 161.47;157.65 160.08 158.02 154.43 158.16 145.79
158.30 176.26;158.46 158.02 159.16 161.30 157.72 142.67 158.01
161.11;157.96 160.11 158.72 177.96 155.15 142.76 157.72
154.23;158.14 160.09 159.12 162.26 168.63 119.31 168.22
162.27]%Vector de tensoes (Entrada)

P=P'%Transposta do vector

T=T'%Transposta do vector

net=newff(P,T,[N1 N2],{'tansig' 'tansig' 'purelin'},'traingdx') %Criação da
rede

net.divideFcn='' %para não dividir os dados em treino, validação e teste

net.trainParam.epochs=5000%Numero de épocas

net=train(net,P,T)%Treinamento da rede

y=sim(net,P)%Simulação da rede com o vector P
Furo4= [157.85 160.95 159.35 169.33 157.14 144.16 159.46 156.63]
%Vector de tensao do Furo4 para testar a rede

Furo4=Furo4'%Transposta do vector

y2=sim(net,Furo4)%Simulação da rede com o vector de tensoes do Furo4
```

CAPÍTULO 6 - RESULTADOS E DISCUSSÃO

6.1 Resultados

Após realizar uma primeira análise com os dados mostrados nas Tabelas 1 e 2 (Seção 3.2.8), pode-se concluir que não eram dados suficientes para treinar a rede neural e obter alguns resultados representativos do programa. Conclui-se que foi melhor manter os dados dos cantos das peças (representados pelo 'x' vermelho na Figura 31), e adicionar novos dados que estarão localizados no meio dos lados da peça (representados pelo 'x' preto na Figura 31). E também, pode-se observar os diferentes tipos de furos numerados de um a cinco que estão representados na Figura 31.

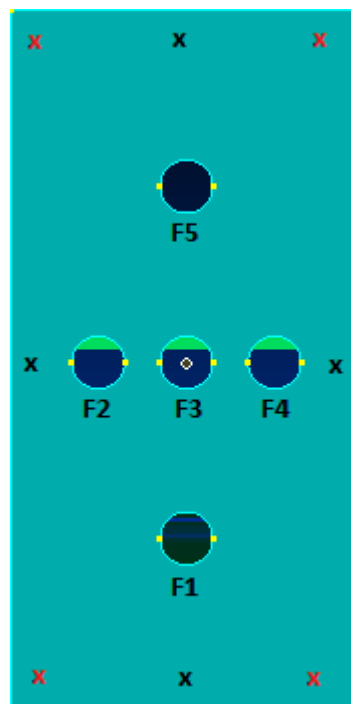


Figura 31 - Novos dados obtidos e furos numerados

Lembrando-se que os dados dos sensores estão numerados começando do canto inferior esquerdo, que corresponde ao valor de X1, para o canto inferior direito que corresponde ao valor de X2, e respectivamente para os cantos superiores com os valores de X3 e X4. Ou seja, a partir do canto inferior esquerdo, andando no sentido anti-horário, a

sequencia de numeração dos pontos ou possíveis sensores são: X1, X1-2, X2, X2-3, X3, X3-4, X4 e X4-1.

Os novos dados das peças estão apresentados nas Tabelas 3 e 4.

Tabela 3 - Novos dados tomados das peças de R=15 mm.

R=15	X1(MPa)	X1-2	X2(MPa)	X2-3	X3(MPa)	X3-4	X4(MPa)	X4-1
P1	162,85	146,14	162,75	160,65	158,53	158,32	148,35	160,07
P2	159,7	160,13	160,9	157,37	158,01	149,02	158,58	168,04
P3	159,89	159,58	159,33	159,02	158,52	145,45	158,44	159,25
P4	157,85	160,95	159,35	169,33	157,14	144,16	159,46	156,63
P5	158,15	159,72	160,99	160,37	163,18	130,33	161,13	159,85

Tabela 4 - Novos dados tomados das peças de R=20 mm.

R=20	X1(MPa)	X1-2	X2(MPa)	X2-3	X3(MPa)	X3-4	X4(MPa)	X4-1
P1	165,75	138,56	167,19	162,58	157,22	143,66	158,63	161,47
P2	157,65	160,08	158,02	154,43	158,16	145,79	158,30	176,26
P3	158,46	158,02	159,16	161,30	157,72	142,67	158,01	161,11
P4	157,96	160,11	158,72	177,96	155,15	142,76	157,72	154,23
P5	158,14	160,09	159,12	162,26	168,63	119,31	168,22	162,27

Estes novos dados podem ser introduzidos no programa MATLAB[®] das redes neurais e serão obtidos resultados mais representativos para o problema apresentado.

O processo de análise consistiu em testar se a rede é capaz de determinar a posição de cada furo para um número determinado de épocas com máximo de 5000 épocas, e para um número máximo de duas camadas ocultas, onde na primeira a camada há 50 neurônios e na segunda 20 ou 10. Tentou-se escolher entre os dois tamanhos de raio para realizar as análises.

Para a interpretação das Figuras 32 a 36, sabe-se que o furo está representado pelos círculos azuis em linha contínua, e a resposta do programa pelo furo em linhas pretas semi-contínuas. Nestas Figuras 32 a 36, os furos estão representados como elipses devido à representação das escalas.

Furo 1

Dados esperados: $R=20$ mm, $X=0$ e $Y= - 100$ mm.

Comparação da análise do Furo 1 para a análise com 5000 épocas e 50 neurônios na primeira camada, e 10 neurônios (Figura 32a) e 20 neurônios (Figura 32b).

$R=15,53$ mm

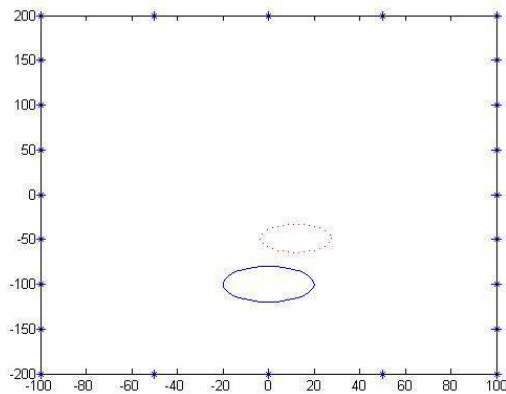
$X=11,90$ mm

$Y= -49,9$ mm

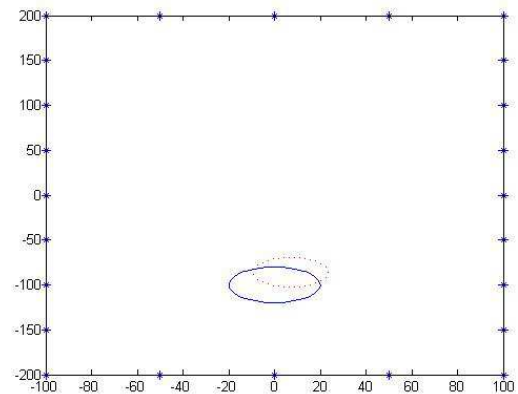
$R=16,32$ mm

$X=7,15$ mm

$Y= -85,81$ mm



(a)



(b)

Figura 32 - Teste para o Furo 1 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda

Pela Figura 32 verifica-se que o resultado para o Furo 1 foi melhor para o segundo caso, onde 20 neurônios foram utilizados na segunda camada intermediária. A Tabela 5 apresenta os erros calculados para cada resultado com relação ao Furo 1

Tabela 5 - Dados dos erros para o Furo 1

10 neurônios na segunda camada				20 neurônios na segunda camada		
Reais (mm)	Dados (mm)	Erro	Erro (%)	Dados	Erro	Erro (%)
R=20	15,53	-0,2235	-22,35	16,32	-0,184	-18,4
X=0	11,9	0	0	7,5	0	0
Y=-100	-49,9	-0,501	-50,1	-85,81	-0,1419	-14,19
Diferença entre os centros (mm): 51,49					16,05	

Furo 2

Dados esperados: R=20 mm, X= - 50 mm e Y= 0 mm.

Comparação da análise do Furo 2 para a análise com 5000 épocas e 50 neurônios na primeira camada, e 10 neurônios (Figura 33a) e 20 neurônios (Figura 33b).

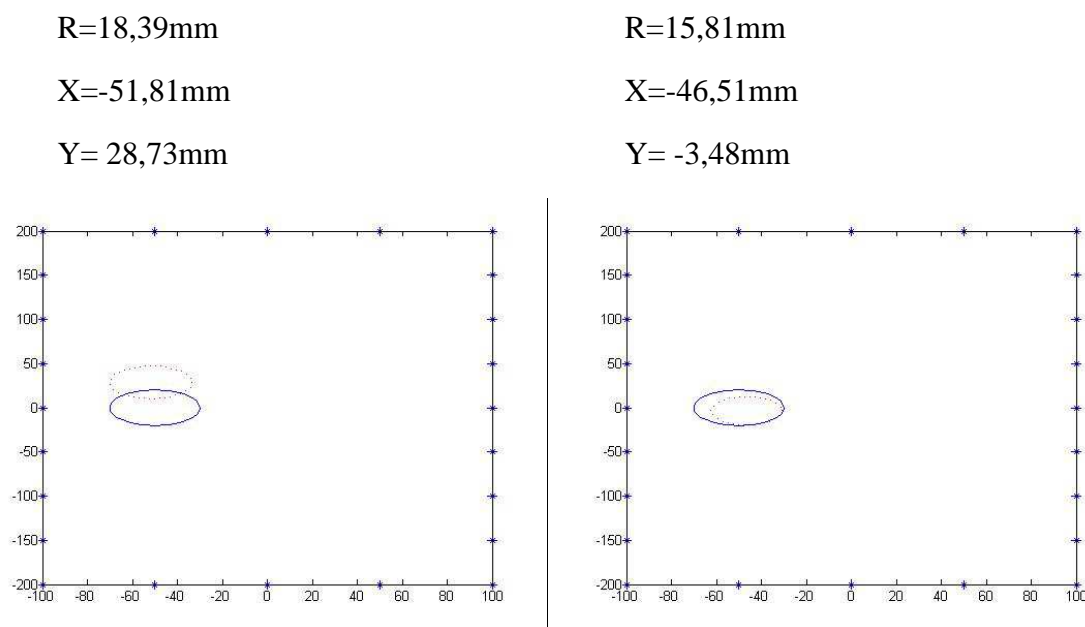


Figura 33 - Teste para o Furo 2 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda

Pela Figura 33 verifica-se que o resultado para o Furo 2 também foi melhor para o segundo caso, onde 20 neurônios foram utilizados na segunda camada intermediária. Na seguinte tabela se apresentam os valores dos erros.

Tabela 6 - Dados dos erros para o Furo 2

10 neurônios na segunda camada				20 neurônios na segunda camada		
Reais (mm)	Dados (mm)	Erro	Erro (%)	Dados	Erro	Erro (%)
R=20	18,39	-0,0805	-8,05	15,81	-0,2095	-20,95
X=-50	-51,81	0,0362	3,62	-46,51	-0,0698	-6,98
Y=0	28,73	0	0	-3,48	0	0
Diferença entre os centros (mm): 28,78					4,92	

Furo 3

Dados esperados: R=15 mm, X= 0 mm e Y= 0 mm.

Comparação da análise do Furo 3 para a análise com 5000 épocas e 50 neurônios na primeira camada, e 10 neurônios na primeira imagem e 20 na segunda.

R=17,47mm

X=4,8mm

Y= 4,8mm

R=16,54mm

X=-7,9mm

Y= 3,52mm

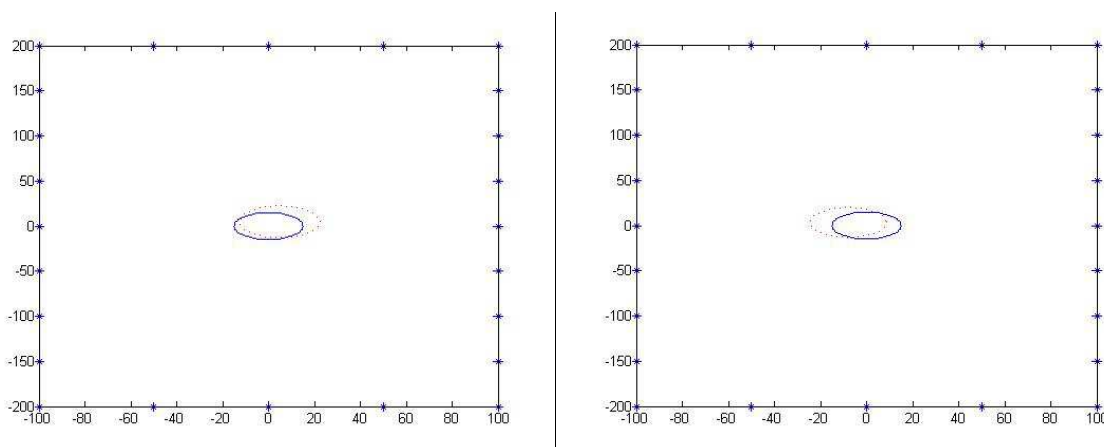


Figura 34 - Teste para o Furo 3 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda

Pela Figura 34 verifica-se que o resultado para o Furo 3 foi semelhante para ambos os casos (primeiro considerando 10 neurônios na camada oculta e segundo considerando 20 neurônios na camada oculta). Na tabela 7 se apresentam os valores dos erros.

Tabela 7 - Dados dos erros para o Furo 3

10 neurônios na segunda camada				20 neurônios na segunda camada		
Reais (mm)	Dados (mm)	Erro	Erro (%)	Dados	Erro	Erro (%)
R=15	17,47	0,164	16,4	16,54	0,102	10,2
X=0	4,8	0	0	-7,9	0	0
Y=0	4,8	0	0	3,52	0	0
Diferença entre os centros (mm): 6,78					8,64	

Furo 4

Dados esperados: R=15 mm, X= 50 mm e Y= 0 mm.

Comparação da análise do Furo 4 para a análise com 5000 épocas e 50 neurônios na primeira camada, e 10 neurônios na primeira imagem e 20 na segunda.

R=18,03mm

X=30,45mm

Y= 1,25mm

R=17,36mm

X=23,15mm

Y= 0,14mm

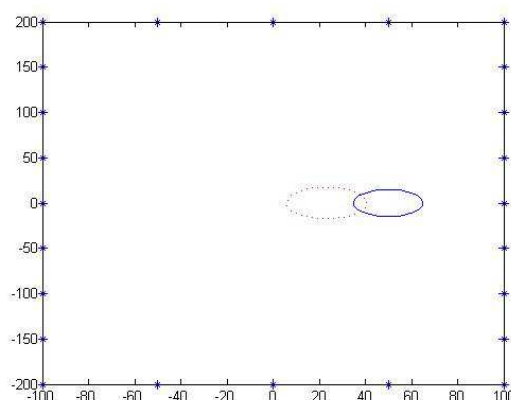
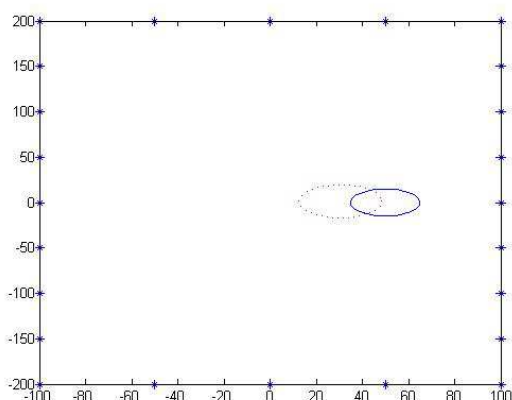


Figura 35 - Teste para o Furo 4 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda

Pela Figura 35 verifica-se que o resultado para o Furo 4 foi um poço melhor para o segundo caso, mas mesmo assim os valores não se ajustam muito ao valor esperado. A Tabela 8 apresenta os erros calculados para cada resultado com relação ao Furo 4.

Tabela 8 - Dados dos erros para o Furo 4

10 neurônios na segunda camada				20 neurônios na segunda camada		
Reais (mm)	Dados (mm)	Erro	Erro (%)	Dados	Erro	Erro (%)
R=15	18,03	0,202	20,2	17,36	0,1573	15,73
X=50	30,45	-0,391	-39,1	23,15	-0,537	-53,7
Y=0	1,25	0	0	0,14	0	0
Diferença entre os centros (mm): 19,58					26,85	

Furo 5

Dados esperados: R=20 mm, X= 0 mm e Y= 100 mm.

Comparação da análise do Furo 5 para a análise com 5000 épocas e 50 neurônios na primeira camada, e 10 neurônios na primeira imagem e 20 na segunda.

R=16,00mm

X=43,05mm

Y= 79,66mm

R=17,91mm

X=.10,85mm

Y= 92,87mm

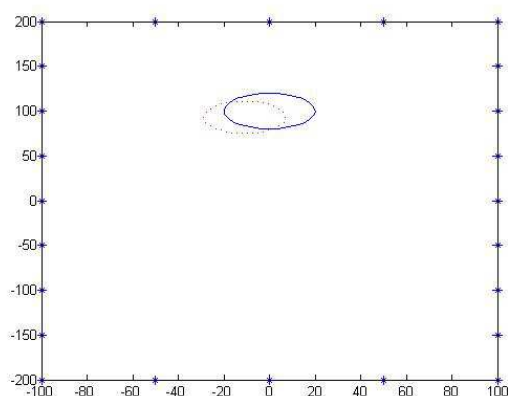
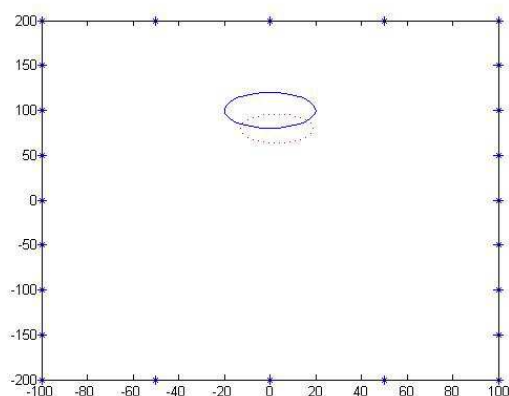


Figura 36 - Teste para o Furo 5 com 50 neurônios na primeira camada e (a) 10 neurônios na segunda; (b) 20 neurônios na segunda

Pela Figura 36 verifica-se que o resultado para o Furo 5 se ajusta muito melhor no caso em que 20 neurônios foram utilizados na segunda camada intermediária, que onde foram utilizados 10. A Tabela 9 apresenta os erros calculados para cada resultado com relação ao Furo 5.

Tabela 9 - Dados dos erros para o Furo 5

10 neurônios na segunda camada				20 neurônios na segunda camada		
Reais (mm)	Dados (mm)	Erro	Erro (%)	Dados	Erro	Erro (%)
R=20	16	-0,2	-20	17,91	-0,1045	-10,45
X=0	43,05	0	0	10,85	0	0
Y=100	79,66	-0,2034	-20,34	92,87	-0,0713	-7,13
Diferença entre os centros (mm): 47,61					12,98	

Foram criadas tabelas com os dados de cada teste e com os erros realizados pelo programa para calcular as diferentes posições dos furos. O tipo de erro utilizado foi o erro relativo que realiza a divisão entre a diferença da posição exata e a dada pelo programa, e a posição exata. Por isso, às vezes, existe uma fila cheia de zeros, já que existem posições exatas de coordenada zero, mas isso não significa que seu erro seja zero. Pode se apreciar nas figuras o local dos diferentes furos com respeito à posição exata. Além disso, foi calculada, por trigonometria, a distância entre os centros do furo achado pelo programa, e do furo real, já que são conhecidas as posições dos centros de cada furo para poder apreciar quanto foi o deslocamento deles.

Como conclusão geral na última coluna da Tabelas pode se apreciar que quando o número de neurônios da segunda camada oculta é aumentado para 20, o erro relativo (em porcentagem) é menor, visto que a resposta do programa é melhor para este caso. A quantidade de neurônios poderia ser aumentada para investigar se o erro diminuiria ainda mais, mas quanto mais neurônios é colocado na implementação do programa, maior é o custo computacional para executar este programa.

CAPÍTULO 7 - CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

7.1 Conclusões

Conclui-se que este trabalho mostra que os resultados obtidos acertam as predições que se tinham sobre o que se queria obter, já que introduzindo os dados de tensões no algoritmo das redes neurais foi capaz de determinar com uma precisão determinada a posição do furo. Analisando os resultados gráficos, as tabelas, pode se tirar a conclusão que neste caso, a posição e o tamanho dos furos se ajustam com maior precisão no caso cuja segunda camada tem 20 neurônios do que na camada com 10 neurônios. Mas, mesmo assim, os resultados não são tão precisos quanto se desejaria já que às vezes ocorre um ajuste melhor para a camada de 10 neurônios. Também não se ajustam perfeitamente a posição e o tamanho dos furos. Pode se concluir que como método de detecção de danos superficiais possibilita perceber onde poderiam estar localizados os furos. Para poder obter melhores predições teria que se dispor de um banco de dados maior. Mas, mesmo assim, pode se afirmar que as redes neurais são uma ferramenta de cálculo muito potente que permitiu a realização deste trabalho.

7.2 Sugestões para trabalhos futuros

Como sugestões para trabalhos futuros, a precisão dos resultados poderia ser melhorada mediante uma tomada maior de dados, que possibilita gerar um banco de dados maior, permitindo obter resultados mais representativos, para que o algoritmo da rede neural tenha maior capacidade de aprendizagem, e seja capaz de determinar com maior exatidão a posição dos furos.

Outras possíveis sugestões para trabalhos futuros poderiam ser a realização de testes de laboratório para poder comparar os resultados obtidos neles com os obtidos nos programas de elementos finitos, e a implementação deles dentro do algoritmo das redes neurais para fazer uma comparação e entre os processos numéricos e os de laboratório.

REFERÊNCIAS BIBLIOGRÁFICAS

- Aluminum 7075 Heat Treated (T6) Wrought data sheet. Disponível em: <http://www.matweb.com/search/SpecificMaterial.asp?bassnum=MA7075T6>. Acesso em: 15 Out 2012
- Clough, R.W. *The Finit Element Method in Plane Stress Analysis*, Proceeding of 2nd ASCE Conference on Electronic Computation, Pittsburgh, PA, September 8-9, 1960.
- Cook, R.D. **Finite element modeling for stress analysis**. New York: John Wiley & Sons, Inc, 1995. 320 p
- Courant, R. *Variational Methods for the Solutions of Problems of Equilibrium and Vibration*, Bulletin of American Mathematical Society, Vol 49, 1943.
- FEMAP – Finite Element Modeling And Postprocessing, Version 10.2.0. Siemens Product Lifecycle Management Software Inc, 2010.
- FEMAP - Real FEA made easy. Disponível em: http://www.plm.automation.siemens.com/en_us/products/velocity/femap/index.shtml. Acesso em: 20 Jan 2013.
- Chen, G.; Dong, X. **‘From Chaos to Order’: Methodologies, Perspectives and Applications**” Wolrd Scientific Series, 1988.
- Greenstadt, J. **On the Reduction of Continuous Problems to Discret Form** IBM J. Res. Dev., Vol 3, 1959.
- Hebb, D.O. **‘Introduction and Chapter 4: The First Stage of Perception: Growth and Assembly’**. Neurocomputing Fundation research. 1989
- Hopfield, J. **‘Neural Network and Physical System with Emergent Collective Computational Abilities’**. Neurocomputing Fundation research. 1989.
- Huebner, K. H. **The Finite Element Method for Engineers**. New York: John Wiley, 1975. 500 p.
- Izaurieta, F.; Saavedra, C. **‘Redees Neuronales Artificiales’**. Departamento de Fisica Universidad Concepcion Chile.

- Lin, C.T.; Lee, G. **‘Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems’**. Prentice-Hall, 1996
- Lizarza, J.T.C. **Método de los Elementos Finitos para Análisis Estructural**. UNICOPIA C.B. San Sebastián – Gipuzkoa, 2008.
- McCulloch, W.; Pitts W. **‘A Logical Calculus of the Ideas Immanent in Nervous Systems’**. Neurocomputing Foundation research. 1987.
- Minsky, M.; Papert, S. **‘Perceptrons’**. Neurocomputing Foundation research. 1989.
- Reddy, J.N. **An Introduction to the Finite Element Method**. 3 ed. New York: McGraw-Hill, 2006. 766 p.
- Rosenblatt, F. **‘The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain’**. Neurocomputing Foundation research. 1989.
- Rumelhart, D.E. Hinton, G.E. **‘Learning Internal Representations by Error Propagation’**. Neurocomputing Foundation research. 1989.
- Ottosen, N.S.; Peterson, H. **Introduction to the Finite Element Method**. New York: Prentice Hall, 1992. 410 p
- Serrano, A.J.; Soria, E.; Martín, J.D. **‘Redes Neuronales Artificiales’**. Universidad Valencia, Espanha. 2009.
- Lopes, P.S. **‘Detecção de danos em estruturas por meio de técnicas de redes neurais artificiais e algoritmos genéticos’**. Dissertação de mestrado (projeto e Fabricação), UNIFEI, Itajubá, 2007.
- Turner, M.J.; Clough, R.W.; Martin, H.C.; Topp, L.C. **Stiffness and Deflection Analysis of Complex Structures**, J. Aeronaut. Sci., Vol. 23, No. 9, 1956.
- Soporte FEMAP & NX Nastran. Disponível em: <<http://www.iberisa.com/soporte.htm>>. Acesso em: 20 Jan 2013.
- Soporte MATLAB. Disponível em: < <http://www.mathworks.com>>. Acesso em: 20 Maio 2013.
- Werbos, P. **‘Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences’**. University press, Cambridge, 1974.
- Zienkiewicz, O.C.; Cheung, Y.K. **Finite Elements in the Solution of Field Problems**, Engineer, Vol. 220, 1956.

APÊNDICE A - Conceito de tensão numa placa com furo

Neste apêndice será discutido sobre o estado tensional de um furo numa placa que está submetida a um carregamento transversal uniforme, pois a realização de furos em placas é uma necessidade habitual no campo da engenharia. Podem ocorrer defeitos deste tipo no material e é importante conhecer certos dados sobre eles. O principal fator é que o furo atua como um concentrador de tensões, isso porque modifica a distribuição dos esforços, surgindo maiores esforços em alguns lugares da peça. Neste caso, o valor da tensão máxima que se produz dentro da peça vai ser três vezes o valor do carregamento lateral. O cálculo desta tensão será explicado abaixo.

A Figura 27 mostra uma placa furada na parte central da peça é submetida a um carregamento lateral de tração uniaxial

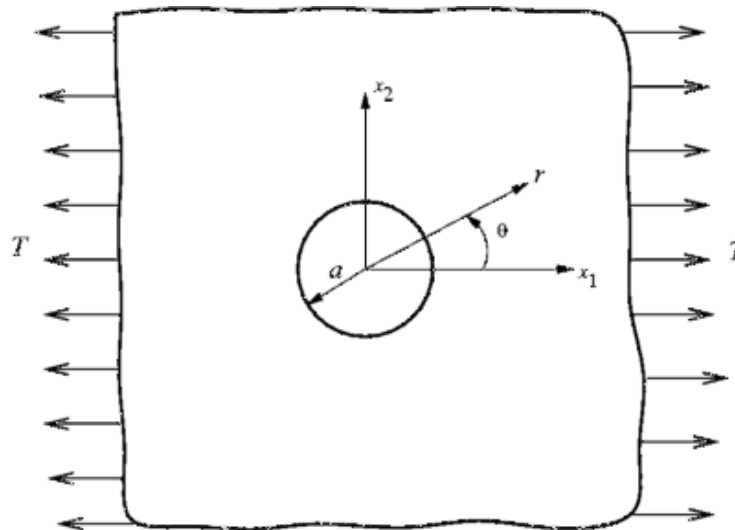


Figura 37 - Esquema de peça furada submetida a tração uniaxial

As equações que regem o estado tensional de uma placa submetida a um carregamento axial com um furo, expressadas em coordenadas polares, são a tensão radial σ_{rr} (Equação(50)), tensão circunferencial $\sigma_{\theta\theta}$ (Equação(51)) e tensão tangencial $\sigma_{r\theta}$ (Equação (52)).

$$\sigma_{rr} = \frac{T}{2} \left(1 - \frac{a^2}{r^2}\right) + \frac{T \cos(2\theta)}{2} \left(\frac{3a^4}{r^4} - \frac{4a^2}{r^2} + 1\right) \quad (50)$$

$$\sigma_{\theta\theta} = \frac{T}{2} \left(1 + \frac{a^2}{r^2}\right) - \frac{T \cos(2\theta)}{2} \left(\frac{3a^4}{r^4} + 1\right) \quad (51)$$

$$\sigma_{r\theta} = \frac{T \sin(2\theta)}{2} \left(\frac{3a^4}{r^4} - \frac{2a^2}{r^2} - 1\right) \quad (52)$$

Com:

a – Radio do furo;

r – Coordenada radial;

T – Tensão uniaxial;

θ – Coordenada angular.

E fácil observar que para o valor de ($r = a$) somente será obtida uma equação que seja diferente de zero, a tensão circunferencial ($\sigma_{\theta\theta}$) (Equação (53)).

$$\sigma_{\theta\theta} = T - 2T \cos(2\theta) \quad (53)$$

Pode-se ver que as equações serão máximas para valores diferentes do ângulo θ , nas Equações (54) e (55)

$$\theta = 0 \rightarrow \sigma_{\theta\theta} = -T \quad (54)$$

$$\theta = \frac{\pi}{2} \rightarrow \sigma_{\theta\theta} = 3T \quad (55)$$

Tomado o valor absoluto da fração (entre a Equação (55) e a Equação (54)) pode-se obter o fator concentrador de tensões que será igual a três, como também três vezes será o que se amplifique a tensão no ponto de tensão máxima. O furo se deformará devido ao efeito das tensões uniformes e virará na forma de uma elipse com seu eixo maior na mesma direção que o carregamento e eixo menor no sentido perpendicular. Os pontos de tensão máxima ficarão nos extremos do eixo menor da suposta elipse. Na Figura 28, pode-se visualizar de uma melhor maneira o exposto acima.

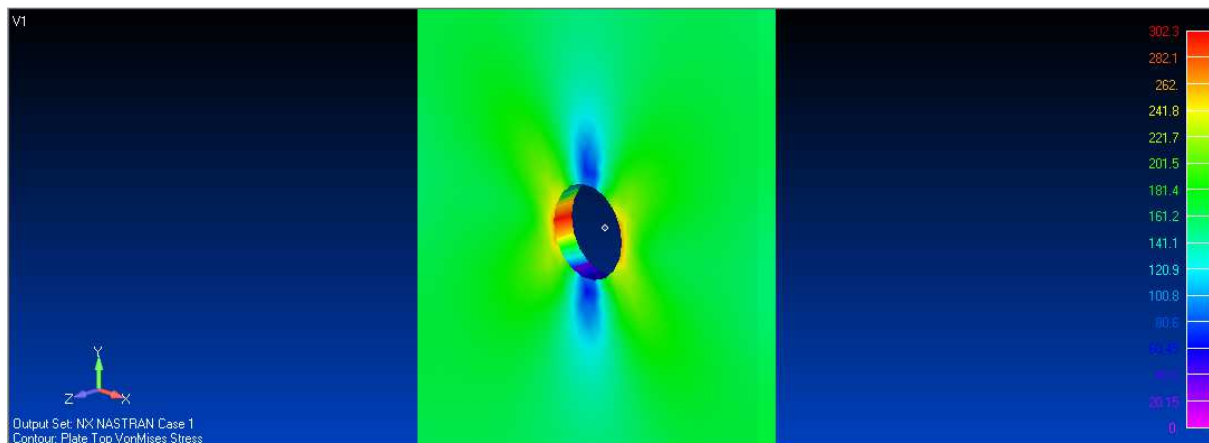


Figura 38 - Zoom no furo

Nesta Figura 28 observa-se a peça já analisada com o resultado das tensões aplicadas, e pode-se perceber como as tensões máximas se localizam nos extremos do furo.

Na Figura 29 pode-se ver como se deforma a peça quando se aplica o carregamento. Observa-se perfeitamente como o furo circular se torna uma elipse, e os pontos de máxima tensão ficam nos extremos do menor eixo.

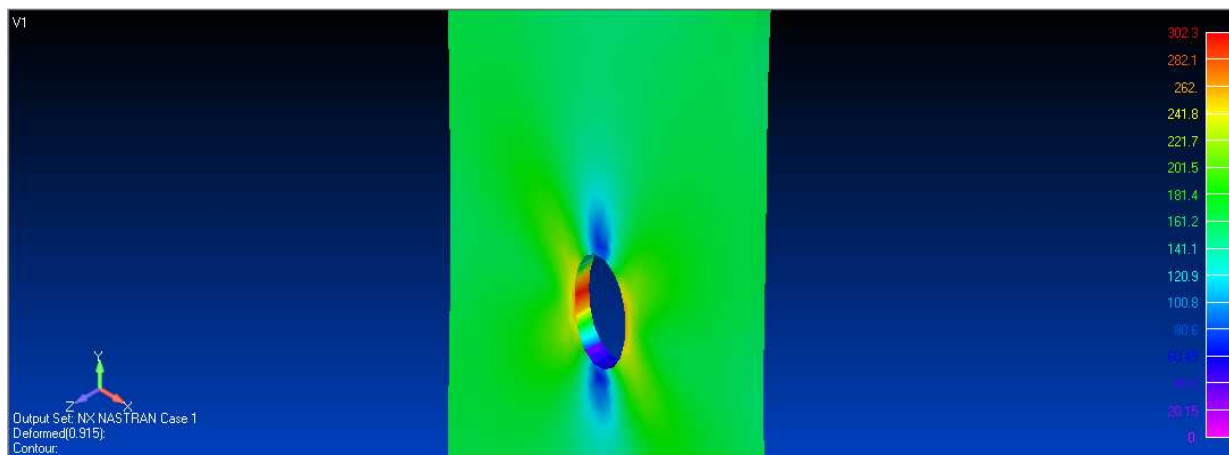


Figura 39 - Zoom no furo deformado

APÊNDICE B - Critério de *Von Misses*

O critério de *Von Misses* fala que o material começará a ceder quando a tensão de *Von Misses* atinge um valor de tensão crítica. A tensão de *Von Misses* é utilizada para prever o escoamento dos materiais que se encontram submetidos a qualquer carregamento que origine as tensões uniaxiais.

A formulação matemática das equações de *Von Misses* pode se expressar mediante as tensões principais, conforme a Equação (56).

$$(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_1 - \sigma_3)^2 = 2\sigma_v^2 \quad (56)$$

Esta Equação (56) pode ser expressa de este outro modo, como na Equação (57).

$$(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy} + \tau_{yz} + \tau_{zx})^2 = 2\sigma_v^2 \quad (57)$$

E supondo que a espessura da peça é relativamente pequena em comparação as outras dimensões, pode-se inferir que a peça não terá tensões na direção z , portanto, pode-se aplicar as equações da tensão de *Von Misses* em duas dimensões (Equação (58)).

$$(\sigma_x^2 - \sigma_x \sigma_y + \sigma_y^2 + 3\tau_{xy}^2)^2 = \sigma_v^2 \quad (58)$$

Uma comprovação foi feita no estudo para verificar se os valores obtidos pelas equações de *Von Misses* conferiam com os dados obtidos pelo programa. Os valores dos dados dos pontos de tensão máxima nas direções x , y , junto com o valor da tensão de cisalhamento foram utilizados. Estes valores foram introduzidos na Equação (58), para ver se conferiam com o valor mostrado pela análise do programa.

Na Equação (59) são apresentados os valores que foram obtidos pela análise do programa, depois estes valores foram substituídos na Equação (58) para ver se conferiam com o valor dado pela análise de *Von Misses* no programa FEMAP/NASTRAN® ($\sigma_v = 252.11 \text{MPa}$), como pode ser visto na Figura 30. Os dados das tensões normais e da tensão de cisalhamento também foram tomados no mesmo elemento onde foi feita a análise de *Von Misses*.

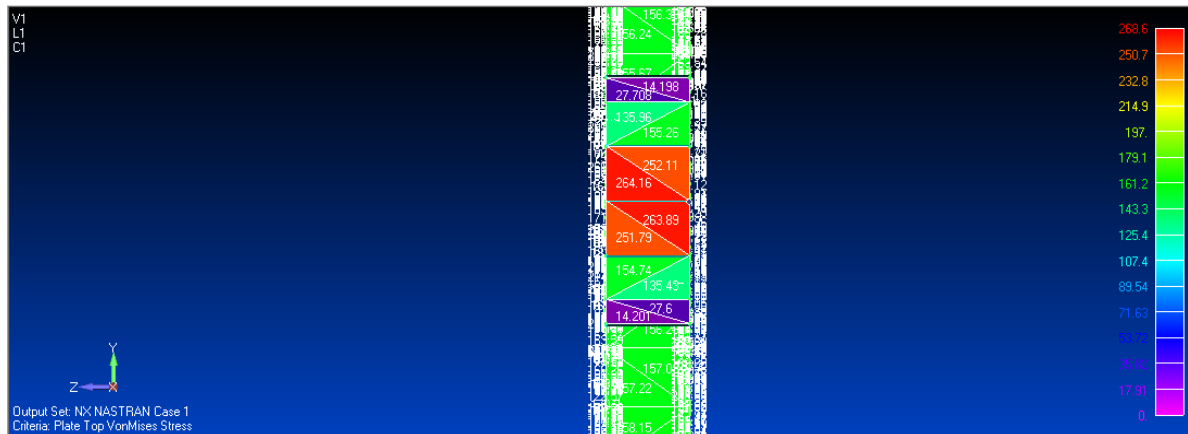


Figura 40 - Zoom dados de teste de VonMisses

Os valores dos dados tomados foram os seguintes.

$$\begin{aligned}
 \sigma_x &= 263,65 \text{ MPa} \\
 \sigma_y &= 25,13 \text{ MPa} \\
 \tau_{xy} &= 3,72
 \end{aligned}
 \tag{59}$$

Quando estes valores da Equação (59) foram substituídos na Equação (58), foi observado que o resultado numérico obtido foi o valor de $\sigma_v = 252,1 \text{ MPa}$ que é aproximadamente igual ao valor proporcionado pelo software.